VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Bezpečné propojení SIP serverů

Secure peering of SIP servers

2010 **Bc. Zbyněk Kurial**

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 7.5.2010 ..……

Bc. Zbyněk Kurial

Poděkování

Děkuji vedoucímu mé diplomové práce doc. Ing. Miroslavu Vozňákovi, Ph.D., za cenné rady, podporu a odborné vedení při sestavování této práce.

Abstrakt

Tato diplomová práce se zabývá využitím zabezpečovacích mechanismů SIP protokolu. Definuje pojmy a zásady pro technické zabezpečení tohoto protokolu v rámci internetové telefonie. Jsou popsány a vysvětleny pojmy SRTP, IPsec, VPN, TLS a mechanismy souvisejícími s těmito pojmy. Praktická část se zabývá návrhem virtuálních tunelů realizovaných projekty OpenSWAN a OpenVPN a aplikování těchto mechanismů na VoIP telefonii. Poslední část této práce je realizace zabezpečení SIP komunikace pomocí TLS a použití SRTP protokolu pro šifrování hovorových dat. Bezpečnostní aplikace jsou zaměřeny na komunikaci mezi Asterisk servery v nezabezpečené síti simulující internet.

Klíčová slova

VoIP, RTP, SRTP, SIP, Asterisk, IPsec, VPN, TLS, OpenVPN, OpenSWAN, SIPS

Abstract

This diploma thesis deals with the use of security mechanisms for SIP protokol. It defines the concepts and principles for the protocol technical support in the context of Internet telephony. It describes and explains concepts of the SRTP, IPsec, VPN, TLS and relating mechanisms. The practical part deals with design of virtual tunnels realized by OpenSWAN and OpenVPN and with applying these mechanisms to VoIP telephony. The last part of this diploma thesis is the realization of security comunication of SIP messages by TLS and application of SRTP protocol for the encryption of call data. The secure aplications are focused on communication between Asterisk servers in an unsecured network simulating the Internet.

Key words

VoIP, RTP, SRTP, SIP, Asterisk, IPsec, VPN, TLS, OpenVPN, OpenSWAN, SIPS

Seznam použitých symbolů a zkratek

AES **Advanced Encryption Standard** - šifrovací mechanismus
AH **Authentication Header** - protokol pro zapouzdření dat u IPsec
CA **certificate authority** - certifikační autorita
DH **Diffie-Hellman** - algoritmus pro výměnu klíčů
DES **Data Encryption Standard** - šifrovací mechanismus
ESP **Encapsulating Security Payload** - protokol pro zapouzdření dat u IPsec
GRE **Generic Routing Encapsulation** - protokol pro zabezpečení IP protokolu
G.711 hlasový kodek
G.729 hlasový kodek
GSM **Global System for Mobile communications (Groupe Spécial Mobile)** - standard pro mobilní komunikaci
H.323 signalizační protokol ve VoIP
HMAC **Hash-based Message Authentication Code** - hashovací algoritmus
HTTP **Hypertext Transfer Protocol** - textově internetovaný protokol
IAX **Inter-Asterisk eXchange** - protokol pro propojení VoIP serverů
ICV **Integrity Check Value** - pole v AH záhlaví
IKE **Internet Key Exchange** - protokol pro výměnu klíčů v IPsec
IP **Internet protokl** - protokol pro přenos dat v paketové síti
IPsec **IP security** - protokol pro zabezpečení IP protokolu
ISAKMP **Internet Security Association and Key Management Protocol** - protokol pro vyjednání bezpečnostní asociace u IPsec
L2TP **Layer 2 Tunneling Protocol** - protokol pro vytvoření bezpečného tunelu
MD5 **Message-Digest algorithm 5** - hashovací funkce
MITM **man-in-the-middle** - útok na IP síť
MGCP **Media Gateway Control Protocol** - signalizační protokol ve VoIP
NAT **Network Address Translation** - překlad síťových adres
PCM **Pulse-code modulation** - pulzně kódová modulace
PGP **Pretty Good Privacy** - šifrovací algoritmus
PKCS **Public-Key Cryptography Standards** - soubor kryptografických standardů
PPTP **Point-to-Point Tunneling Protocol** - protokol pro vytvoření bezpečného tunelu
PSK **pre-shared key** - sdílený bezpečnostní klíč
PSTN **Public Switched Telephone Network** - veřejná telefonní síť
QoS **Quality of Service** - kvalita služby
RR **Receiver Report** - soubor statistik od příjemců v RTCP
RSA **Rivest, Shamir, Adleman** - veřejný šifrovací klíč
RTP **Real-time transport Protocol** - protokol pro přenos hlasu
RTCP **RTP Control Protocol** - rozšíření RTP protokolu
S/MIME **Secure/Multipurpose Internet Mail Extensions** - metoda pro autentizaci v elektronické poště a SIP
SA **Security Asociantion** - bezpečnostní asociace u IPsec

SDP **Session Description Protocol**- protokol ve VoIP
SHA **Secure Hash Algorithm** - hashovací funkce
SIP **Session Initiation Protocol** - signalizační protokol ve VoIP
SIPS **SIP Security** - zabezpečení SIP protokolu
SMTP **Simple Mail Transfer Protocol -** protokol pro přenos elektronické pošty
SPI **Security Parameter Index-** index bezpečnostní asociace u IPsec
SR **Sender Report** - soubor statistik u RTCP
SRTP **Secure Real-time Transport Protocol -** zabezpečení RTP protokolu
SSL **Secure Sockets Layer -** zabezpečovací protokol
TAP virtuální rozhranní na spojové vrstvě
TCP **Transmission Control Protocol -** přenosový protokol
TLS **Transport Layer Security -** zabezpečovací protokol
TUN virtuální rozhranní na síťové vrstvě
UA **User Agent** - koncové zařízení v SIP protokolu
UAC **User Agent Client** - koncové zařízení pro vysílání požadavků
UAS **User Agent Server -** koncové zařízení pro vysílání odpovědí
UDP **User Datagram Protocol** - přenosový protokol
VOFR **Voice over Frame Relay** - protokol pro přenos hlasu přes frame relay síť
VOIP **Voice over Internet Protokol** - přenos hlasu přes IP síť
VPN **Virtual Private Network** - virtuální soukromá síť
X.509 specifikační formát uživatelských certifikátů
ZRTP **Zimmerman RTP** - bezpečnostní doplnění SRTP protokolu

Obsah

1 Úvod ... 1
2 IP telefonie ... 2
2.1 VoIP ... 2
2.2 RTP ... 3
2.3 H.323 ... 3
2.4 SIP ... 4
2.4.1 Architektura SIP protokolu ... 4
2.4.2 SIP žádosti a odpovědi ... 5
2.4.3 Registrace k SIP proxy ... 7
2.4.4 Zprostředkování hovoru ... 8
3 Zabezpečení médií v IP telefonii ... 9
3.1 SRTP (secure real-time transport protocol) ... 9
3.2 ZRTP ... 11
3.2.1 Popis ZRTP ... 11
3.2.2 Funkce ZRTP ... 11
3.3 TLS ... 12
3.3.1 Popis TLS ... 12
3.3.2 Požadavky TLS protokolu ... 13
3.3.3 Rozdělení ... 13
3.3.4 Change Cipher Spec Protocol ... 13
3.3.5 Alert protocol ... 14
3.3.6 Handshake protokol ... 15
3.3.7 Hello zprávy ... 15
3.3.8 Record protocol ... 17
3.4 VPN ... 18
3.4.1 Módy VPN ... 19
3.4.2 Požadavky na VPN síť ... 20
3.4.3 Topologie VPN ... 20

3.5 Ipsec 21
3.5.1 AH protokol 22
3.5.2 ESP protokol 23
3.5.3 IPsec SA 24
3.5.4 ISAKMP 24
3.5.5 IKE 24
4 Metody autentizace v protokolu SIP 25
4.1 HTTP Basic Authentication 26
4.2 HTTP Digest Authentication 26
4.3 S/MIME 27
4.4 SIPS 28
5 Realizace zabezpečení SIP serverů 29
5.1 Topologie sítě 30
5.2 Základní nastavení počítačů 31
5.3 Asterisk 31
5.3.1 Konfigurace Asterisku 31
5.3.2 Asterisk Trunk 32
5.4 OpenVPN 33
5.4.1 Konfigurace OpenVPN se sdíleným klíčem 34
5.4.2 Úprava nastavení Asterisku v sip.conf 35
5.4.3 Nastavení routingu 35
5.4.4 Generování TLS certifikátů 35
5.4.5 Konfigurace OpenVPN s TLS 36
5.5 OpenSWAN 37
5.5.1 Úprava nastavení Asterisku v sip.conf 38
5.5.2 Konfigurace pomocí sdíleného hesla 38
5.5.3 Konfigurace pomocí RSA klíčů 39
5.5.4 Konfigurace pomocí TLS 39
5.6 Zabezpečení signalizace SIPS a médií SRTP 40

5.6.1 Zabezpečení signalizace SIPS 41
5.6.2 Certifikáty pro Asterisk 1.6 41
5.6.3 Zabezpečení médií SRTP 43
6 Porovnání použitých řešení 44
6.1 Obtížnost implementace 44
6.2 Odolnost proti útoku 44
6.3 Nárůst datového toku mezi servery 45
7 Závěr 47
Literatura 48
Přílohy 51

1 Úvod

V moderním světě se již telefonování stalo nedílnou součástí každodenního života. Na telefonní komunikaci dnes stojí správný chod nespočet institucí a firem po celém světě. Také v mnoha domácnostech si již nedovedou představit život bez telefonního přístroje. Komunikace se stala globální záležitostí moderního životního stylu. Dnes již po celém světě bezproblémově funguje několik osvědčených telefonních technologií, které jsou schopny globálně zajistit přenos hlasu odkudkoli, kamkoli. Nabízí se tedy otázka, proč je nutno vyvíjet další technologie pro přenos hlasu. Odpověď je velice jednoduchá. Chceme, aby telefonování bylo stále levnější, dostupnější a komfortnější než doposud. Všechny tyto aspekty dokáže zajistit IP telefonie. Jeden z hlavních důvodů vzniku IP telefonie je tedy ekonomika telefonování. VoIP nepotřebuje nákladnou výstavbu mobilních sítí, BTS stanic, antén a vysílačů, nepotřebuje budovat obrovské telefonní ústředny pro zajištění svého provozu. Ještě před započetím vývoje této technologie již měl VoIP svou síť dávno vybudovanou. Právě přenos přes IP síť je nesmírnou ekonomickou výhodou. Nejjednodušší varianta sestavení funkčního VoIP systému jsou dva síťově propojeny počítače s potřebnou softwarovou výbavou. Pokud zprovozníme VoIP pouze v rámci intranetu, neplatíme žádné poplatky a telefonování po internetu s využitím VoIP operátora je také mnohem levnější, než například volání zprostředkované mobilním operátorem.

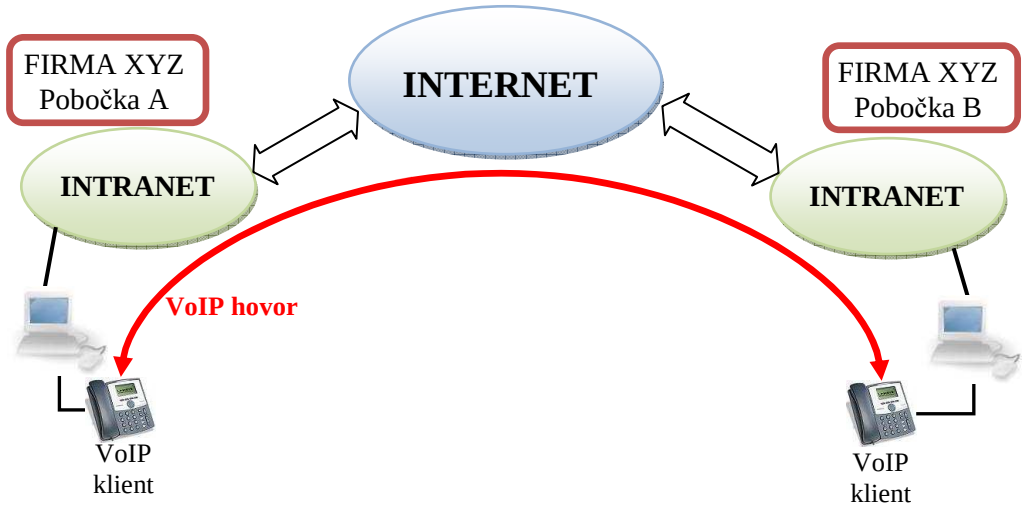
Toto řešení ale přináší řadu problémů spojených s bezpečností provozu. Protože VoIP telefonie je provozovaná přes IP síť, sdílí s ní také všechny slabiny, které ji můžou ochromit. Útok na IP telefonii může u menší firmy v nejhorším případě skončit i krachem. Útočníci mohou po napadení zprostředkovat tisíce telefonních hovorů a následné náklady se mohou vyšplhat do milionů korun.

Právě zabezpečením IP telefonie se budu v diplomové práci zabývat. Popíšu zde základní principy fungování VoIP telefonie na základě SIP protokolu, metody autentizace při registraci k SIP serveru a v praktické části bude několika způsoby vytvořeno bezpečné spojení dvou SIP serverů, které bude tvořit softwarová ústředna Asterisk. Komunikace mezi servery bude probíhat přes vytvořené virtuální tunely, které budou zajištěny SSL/TLS technologií a kvalitním šifrováním AES. Dále bude zabezpečena samotná SIP signalizace pomocí TLS a hovorová data protokolem SRTP. Všechny metody by měly hovoru zajistit maximální bezpečnost před útokem z internetu. Funkčnost těchto VPN tunelů a dalších zabezpečení bude podrobně vysvětlen v teoretické části diplomové práce.

2 IP telefonie

2.1 VoIP

VoIP neboli Voice over Internet Protocol [1, 3, 4, 5] je technologie umožňující digitální přenos hlasu a signalizačních zpráv přes IP síť. Pro přenos se využívá UDP protokol. Nutností k zprostředkování srozumitelného hovoru po IP [2] síti je zajištění několika minimálních technických požadavků. Je důležité hlídat přenosové zpoždění mezi datovými rámci, které by nemělo být vyšší než 100 ms a zajistit určitou minimální přenosovou rychlost linky. Přenos dat po IP síti probíhá digitálně a proto je nutný převod zaznamenaného analogového signálu z mikrofonu do digitální podoby. To mají na starost kodeky vyvinuté speciálně pro technologii VoIP. Je jich na výběr několik a zvolení toho správného závisí na požadované kvalitě hovoru, maximální přenosové rychlosti atd.



Obrázek 1 Spojení hovoru přes internet VoIP technologií

VoIP ovšem také není dokonalým systémem pro telefonní komunikaci. Zásadní nevýhodou této technologie je závislost na funkčnosti samotného internetu. Stačí výpadek serverů internetového providera a už se nikam nedovoláme. Další nevýhodou je bezpečnost systému. Platí zde všechny hrozby, které dnes existují v internetové síti. Spojení hovoru přes internetovou síť je zobrazeno na obrázku 1. A právě zajištění bezpečného provozu, bez možnosti odposlechu a jakékoli manipulace s hovorem je nutností pro rozvoj VoIPu v globálním měřítku.

2.2 RTP

Důležitým protokolem pro přenos hlasu po IP síti je RTP (Real - time Transport Protocol). Byl vyvinut v roce 1996 jako standart RFC 1889 a v roce 2003 vylepšený RFC 3550 [24]. Je nezbytným základem IP telefonie. Důležitou vlastností RTP protokolu je seřazování (sequence number) vysílaných paketů a časové značkování (timestamp) potřebných ke správnému sestavení zvukového záznamu na přijímací straně.

Proces převodu záznamu až po vyslání na přenosové medium proběhne následovně:

- Převod záznamu z mikrofonu do digitální podoby modulací PCM
- Zpracování RTP protokolem, přidání RTP a UDP hlavičky
- Přidání IP hlavičky s logickými adresami odesilatele a příjemce
- Rámec obsahující fyzické adresy a kontrolní součet
- Odeslání na přenosové medium

K přenosu hovorových dat po IP síti může být použit vyspělejší SRTP protokol a ZRTP, které zabezpečí data proti různým útokům. Těmto protokolům se budu věnovat v samostatné kapitole.

Doplněním k RTP je protokol RTCP, který má za úkol poskytovat řídící informace pro tok dat. Přenáší kontrolní pakety pro zpětnou vazbu na zajištění kvality služeb QoS. Zaznamenává důležité údaje o množství odeslaných bajtů, paketů, počet ztracených paketů, kolísání zpoždění, zpětnou vazbu a dobu odezvy. Toto zajišťuje pěti typy zpráv SR (Sender Report), RR (Receiver Report), SDES (Source DEScription), BYE (End message) a APP.

2.3 H.323

Prvním standardem popisující hlasovou komunikaci v reálném čase v IP sítích byl H.323. Vyvinula ho v roce 1996 firma VocalTech, která byla první na světě, nabízející tuto službu. V průběhu další let byl H.323 neustále vyvíjen a vylepšován o bezpečnostní prvky, zajišťující kvalitu služby a další možnosti provozu. V současné době je H.323 již ve verzi 7 z roku 2009.

Protokol H.323 jsem zde uvedl jen okrajově, spíše z důvodu poukázání, že níže uvedený protokol SIP není jediný protokol schopný zajistit komunikaci po IP sítích. H.323 již zde nebudu blíže rozebírat.

2.4 SIP

SIP (Session Initiation Protocol) [8, 9, 10] byl vyvíjen s důrazem na jednoduchost, přehlednost, rozšiřitelnost a flexibilitu. Bylo to reakcí na starší protokol H.323, který byl po létech vývoje příliš rozsáhlý a nepřehledný. SIP byl uveden na trh v roce 1999 jako standard RFC 2543. Ten také prošel vývojem a dnes je jádro protokolu popsáno v RFC 3261 [26].

SIP zajišťuje přenos signalizace telefonního hovoru v IP síti, komunikuje na portu 5060 přes UDP. Hovorová data jsou pak přenášena RTP protokolem. Pracuje na aplikační vrstvě, je textově orientovaný a vychází z protokolů HTTP a SMTP, které jsou v internetu nejpoužívanější. Komunikace je typu server – klient a probíhá formou požadavků a odpovědí tak jak je tomu u HTTP. Na rozdíl od PSTN telefonních sítí je SIP orientován jako end-to-end protokol, logika je uložena v koncových zařízeních, což přináší vyšší výkonnost a odolnost proti chybám.

2.4.1 Architektura SIP protokolu

Základní prvky SIP sítě tvoří UA (user agents) a server, kterému jsou UA přihlášeni. Roli UA může zastávat softwarový klient (SJphone, X-lite, Kphone a další) nebo hardwarový SIP telefon. SIP server je dále rozdělen na několik funkčních částí, z nichž každá vykonává svou danou funkci.

- **UA – user agent**
  - **UAC** – stará se o vysílání požadavků a příjímání odpovědí na něj
  - **UAS** – má opačnou funkci: příjímání požadavků a odpovídání na něj
- **SIP server**
  - **SIP proxy server** – zprostředkování horou
  - **Redirect** – přesměrování
  - **Registrar** – registrace uživatele
  - **Location** – spravuje informace o umístění uživatele

User agent klient komunikuje s User agent server pomocí šesti požadavků obsahující informace o typu požadavku, protokol, port, IP adresu UAS, kterému je žádost zasílána. UAS tyto požadavky přijímá a posílá zpět odpovědi, přičemž na jeden požadavek může být odesláno více odpovědí. Komunikace mezi UAC a UAS je typu peer to peer.

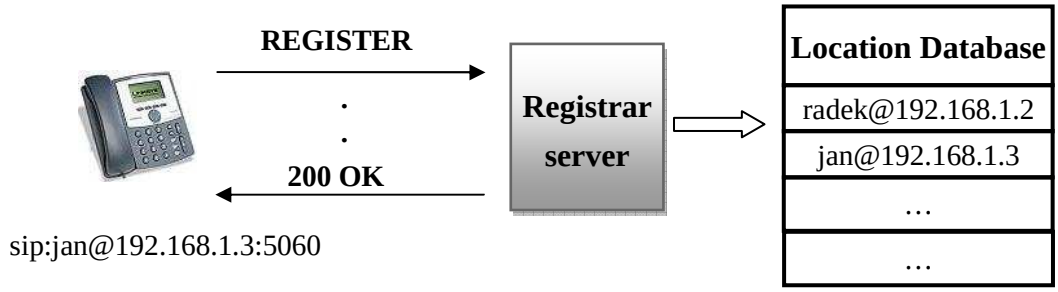
SIP server je velice důležitým prvkem SIP infrastruktury. Stará se o veškeré služby pro zprostředkování telefonního hovoru. Zajistí databázi uživatelů, jejich spravování, vyhledání

volaných účastníků, navazování spojení mezi nimi, stará se o sestavení hovoru, jeho řízení a ukončení.

SIP proxy server je možné rozdělit na dva typy. **Stateless a Stateful server**. Jednodušší a rychlejší je Stateless server. Ten si nezapamatovává stavové informace a směrování není tak pokročilé jako u Stateful serveru, který si vytváří záznamy stavu, které si pamatuje, dokud nedojde k ukončení prováděné akce. To mu dává širší možnosti využití, ale za cenu nižšího výkonu, jelikož musí některé transakce držet poměrně dlouho dobu, například při vyzvánění než protistrana zvedne sluchátko a přijme hovor.

Redirect server se v síti stará o přesměrování, což dovoluje změnit polohu klienta (zaregistrovat se k serveru z jiného místa) a přitom nezměnit svou identitu (zachovat své telefonní číslo). Například pokud se budu v budoucnu k SIP serveru přihlášen pomocí mobilního přístroje, může nastat situace, že s měnící se geografickou polohou přístroje dojde k zaregistrování k serveru, který přísluší dané lokalitě. I v tomto případě budu mít stále stejné kontaktní číslo.

Registrar server zajišťuje registraci uživatele k SIP proxy serveru. Uživatel vysílá žádost REGISTER, následně Registrar server nahlédne do své databáze a pokud tam uživatele najde a souhlasí přihlašovací údaje, uživatele úspěšně zaregistruje. Činnost registrar serveru je znázorněna na obrázku 2.



Obrázek 2 ukázka činnosti Registrar serveru

2.4.2 SIP žádosti a odpovědi

Komunikace SIPu je podobně jako u HTTP založena na žádostech a odpovědích, kde v hlavičce každé žádosti je požadavek upřesněn (Via, From, To, Contact, adt.). Lze z ní vyčíst kdo zaslal žádost, komu je určena, IP adresa serveru, identifikační údaje a další.

Základní žádosti:

- **IVNITE** – požadavek uživatele o zahájení relace (audio, video, hry) je formulovaná v žádosti INVITE. V těle žádosti je popsán popis relace. Žádost je doručena prostřednictvím proxy serveru. Potvrzení je odesláno odpovědí 200 OK.
- **ACK –** značí potvrzení přijetí například odpovědi 200 OK. Na základě ACK je volaný ujištěn, že byla doručena jeho odpověď 200 OK a může začít hovor
- **BYE –** touto žádostí se ukončí spojení, je potvrzena odpovědí 200 OK
- **CANCEL –** tato žádost slouží ke zrušení předchozí žádosti o sestavení spojení. Jestliže volaný dostane žádost INVITE a nepřeje si uskutečnit hovor, odešle žádost CANCEL a hovor bude, odmítnut. Žádost CANCEL nemá vliv na již potvrzené žádosti odpovědí 200 OK.
- **REGISTER –** uživatel se žádostí zaregistruje a odregistruje k SIP serveru
- **OPTIONS –** získání informací o možnostech volaného bez zprostředkování hovoru. Tuto žádost využívá například SIP proxy server ke zjištění stavu UA

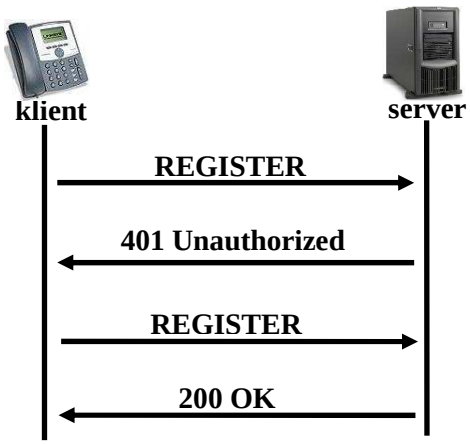
Odpovědi SIP

SIP odpovědí je podstatně více než žádostí, proto jsou rozděleny do šesti kategorií. Každá odpověď má podobu číselného kódu, podle kterého lze určit charakter odpovědi.

- **1XX** – prozatimní nebo také informativní odpověď značící, že server provádí další kroky, na které zatím nemá odpověď. Server pošle 1XX odpověď pokud očekává, že splnění celé žádosti bude trvat déle než 200 ms.
- **2XX** – konečná odpověď například na žádost INVITE. Zaslání odpovědi 200OK značí, že byla žádost akceptována.
- **3XX** – tyto odpovědi poskytují informace o novém umístění uživatele a informace o alternativních službách volání. Jestliže volaný není registrován k proxy serveru, na který přijde žádost o spojení, server vyšle odpověď 3XX obsahující novou jinou polohu volaného.
- **4XX** - tato odpověď značí selhání zpracování žádosti. Problém na straně klienta. Například na žádost REGISTER, která neobsahuje autentizační údaje, vyšle server odpověď 401, ve které klienta o tyto údaje žádá.
- **5XX** – problém na straně serveru, požadavek je v pořádku, ale server pochybil při zpracování.
- **6XX** – odpověď je zaslána v případě, že žádný server není schopen splnit požadavek.

2.4.3 Registrace k SIP proxy

Zaregistrování klienta k SIP proxy zajišťuje Registrar server. Ten spolupracuje s Location database kde jsou uloženi všichni uživatelé, mající povolení užívat daný server. Proces registrace je tvořen žádostí REGISTER, která je po úspěšném zaregistrování potvrzena odpovědí 200 OK. V žádosti REGISTER klient požádá o zaregistrování k serveru, nepošle-li ale registrační údaje, je k tomuto vyzván odpovědí od serveru 401 Unauthorized. V další žádosti REGISTER již klient posílá své registrační údaje. V poli expires lze určit dobu platnosti registrace, jestliže obsahuje nulu, jedná se o žádost o zrušení registrace. Registraci klienta zobrazuje obrázek 3.



Obrázek 3 registrace klienta

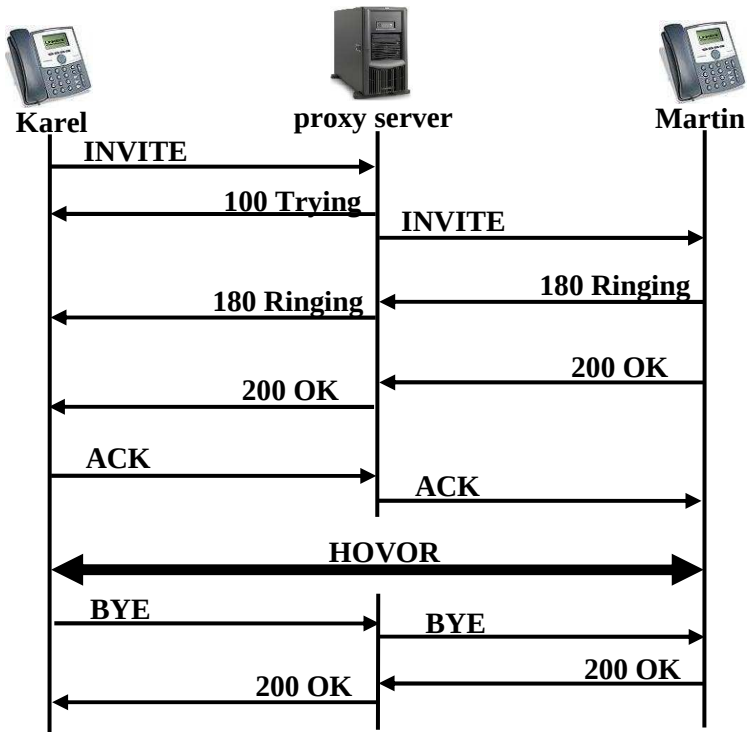
REGISTER sip:128.146.10.103 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.2:18202;branch=z9hG4bK-d87543-341c7771337e1e46-1--d87543-;rport
Max-Forwards: 70
Contact: <sip:radek@192.168.1.2:18202;rinstance=67f3a7ea2745e612>
To: "radek"<sip:radek@192.168.1.1>
From: "radek"<sip:radek@192.168.1.1>;tag=3d417f7f
Call-ID: 33123f0e0f575a20NmFiNTE0MTAxY2E2YTc3NDI2NmZkMGVjMTFlYzhhNjQ.
CSeq: 2 REGISTER
Expires: 3600
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
User-Agent: X-Lite release 1002tx stamp 29712
Authorization: Digest username="radek",realm="asterisk",nonce="2e88ac4e",uri="sip:192.168.1.1",response="c2fb47537aad418c9f5429479745d163",algorithm=MD5
Content-Length: 0

Ze zachycené žádosti o registraci lze vyčíst všechny podstatné informace. Registrační data jsou přehledně rozdělena do několika polí.

- **Via:** ip adresa a port odkud byla vyslaná žádost (192.168.1.2:18202)
- **Contact:** kontakt uživatele (sip:radek@192.168.1.2:18202)
- **To:** odkaz na uživatele v konfiguračním souboru sip.conf – (sip.conf platí pro Asterisk server)
- **Call-ID:** identifikace uživatele
- **Authorization:** Typ autorizace a další přihlašovací údaje, jméno, heslo atd.

2.4.4 Zprostředkování hovoru

Karel zprávou INVITE žádá spojení s Martinem, proxy server mu odpoví opovědí 100 Trying a Martinovi pošle zprávu INVITE. Martin vyšle zprávu 180 Ringing oznamující vyzvánění a při přijetí hovoru vyšle 200 OK. Na to Karel odpoví ACK. Při ukončení hovoru Karlem se vyšle BYE zpráva, která bude Martinem potvrzena v 200 OK. Vše je zobrazeno v obrázku 4.



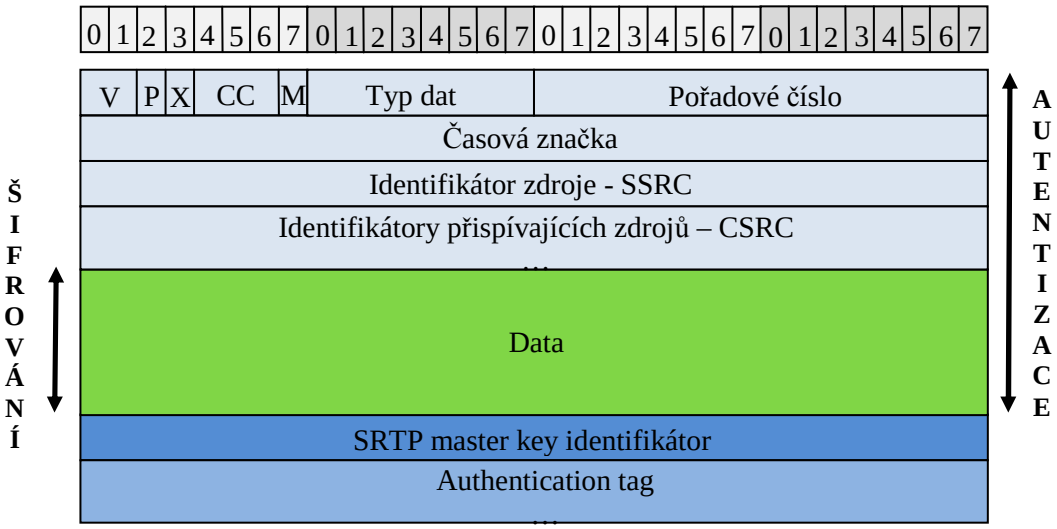
Obrázek č. 4 zprostředkování hovoru

3 Zabezpečení médií v IP telefonii

Přenos hlasových zpráv po IP síti probíhá přes transportní protokol UDP využívající k tomuto navržený RTP protokol. Ten ale nezajišťuje žádné zabezpečení datového provozu a zprávy mohou být lehce zneužity třetí stranou. Běžně dostupné softwarové analyzátory dokážou zachytit a přehrát hlasové zprávy, které jsou nešifrovaně posílány RTP protokolem. Toto je pro komerční využívání technologie VoIP nepřijatelná skutečnost. Proto byl RTP protokol doplněn o zabezpečující prvky, především šifrování dat, autentizace volajících, důvěrnost dat atd. Vznikl SRTP protokol a později ZRTP doplněný o další bezpečnostní prvky.

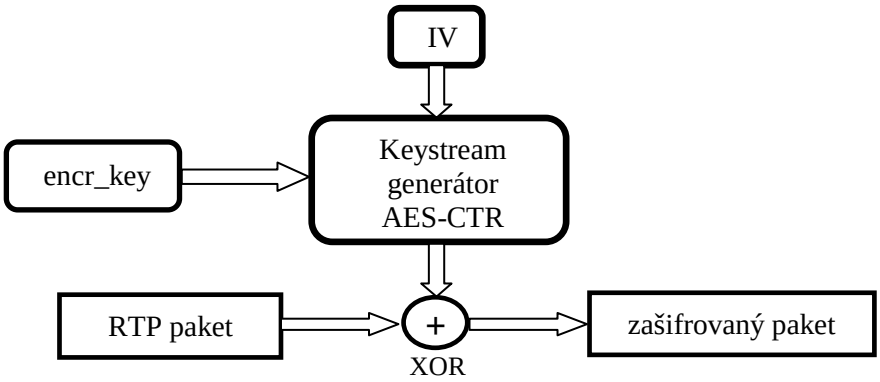
3.1 SRTP (secure real-time transport protocol)

Rozšiřující bezpečnostní zabezpečení jsou integrita přenášených dat, ověřování autenticity, důvěrnost dat a ochrana proti přeposílání. Výchozím portem pro SRTP je port 5004. SRTP [25] je ideální pro ochranu VoIP provozu, protože neovlivňuje přenosové parametry QoS a může být použit ve spojení s kompresí hlaviček paketů. To je důležité při hlasovém hovoru, který je řízen některým z kodeků pro nízký datový tok. SRTP paket je ukázán na obrázku 5.



Obrázek 5 SRTP paket

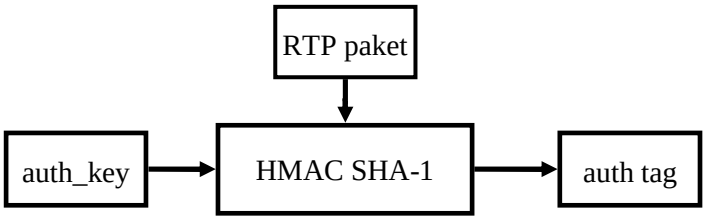
SRTP využívá symetrickou šifru AES [27]. Ta používá sdílené klíče pro šifrování i dešifrování dat o délce 128, 192 a 256 bitů. Data jsou šifrovány postupně v blocích o délce 128 bitů. Nejdříve se vygeneruje klíč, který si oba účastníci vymění a ten je následně aplikován na nešifrovaný obsah paketu pomocí logické operace XOR. Takto zabezpečený paket se pošle příjemci, který ho stejným klíčem dešifruje. Výsledkem je potom původní hlasová zpráva v nezměněné podobě. Zašifrování paketu je ukázáno na obrázku 6.



Obrázek 6 šifrování paketu symetrickým klíčem AES

- IV – 128 bitový inicializační vektor obsahující salt key (náhodné bity + přístupové hesla), SSRC a index paketu
- encr_key - šifrovací klíč

Autentizace je zajištěna hashovací funkcí. Vedle přenášených dat se pošle také vypočtený hash o definované délce podle předem dohodnutého klíče. Takto zašifrovanou zprávu již nejde dešifrovat, protože se jedná o jednosměrné šifrování. Odesilatel zahashuje data sdíleným klíčem a společně s původní podobou zprávy pošle příjemci. Ten nepozměněnou zprávu taky zahashuje a výsledek porovná s příchozím hashem, pokud se shodují, znamená to pro příjemce, že zprávy nejsou nikým pozměněny a jsou zaslány ověřeným účastníkem hovoru. SRTP používá hashování HMAC-SHA-1 zobrazené na obrázku 7. Ten provádí součet hlavičky a datové části paketu a následně ho uloží do pole authentication tag.



Obrázek 7 - hashování funkcí SHA-1

- auth_key – autentizační klíč
- auth tag – zahashovaná zpráva v poli authentication tag

Problémem je ale výměna šifrovacího klíče mezi komunikujícími stranami. Pokud by tato operace byla zachycena útočníkem, tak už by nemohla být zajištěna bezpečnost hovoru. Pro generování sdílených klíčů se používá tzv. master klíč o délce 128, 192 nebo 256 bitů. Ten se ale posílá SDP protokolem, který proti útokům není chráněn.

3.2 ZRTP

3.2.1 Popis ZRTP

ZRTP navrhl světově uznávaný odborník v oblasti zabezpečení datového provozu Philip Zimmermann jako nástavbu k SRTP. Ten již v roce 1990 přišel s protokolem PGP (pretty good privacy), který zajišťuje bezpečný přenos dat bez možnosti odposlouchávání po elektronických kanálech, za což byl vyšetřován americkými bezpečnostními složkami, které požadovaly, aby každý zabezpečovací program měl tzv. zadní vrátka, což by mělo umožnit číst i zašifrované zprávy. Toto ale Philip Zimmerman odmítal s odvoláním na znění americké ústavy o právu na soukromí. To byl hlavní důvod, proč PGP napsal.

Přenos šifrovacího master klíče přes SDP není u SRTP zabezpečen, proto se na tento problém Zimmerman zaměřil. ZRTP nabízí bezpečné mechanismy předání šifrovacích klíčů a ochranu hovoru proti útoku typu Man in the middle. Pro výměnu klíčů je v ZRTP použit Diffie-Hellmanův algoritmus. Jeho úkolem je vytvořit a bezpečně zaslat sdílený tajný klíč, nutný k symetrickému šifrování zpráv. Je založen na umocňování čísel, které si každá strana sama zvolí. Algoritmus zahashuje tajné hodnoty a krátký autentizační řetězec. Vypočtená hodnota se použije pouze jednou, avšak část se uloží pro budoucí spojení.

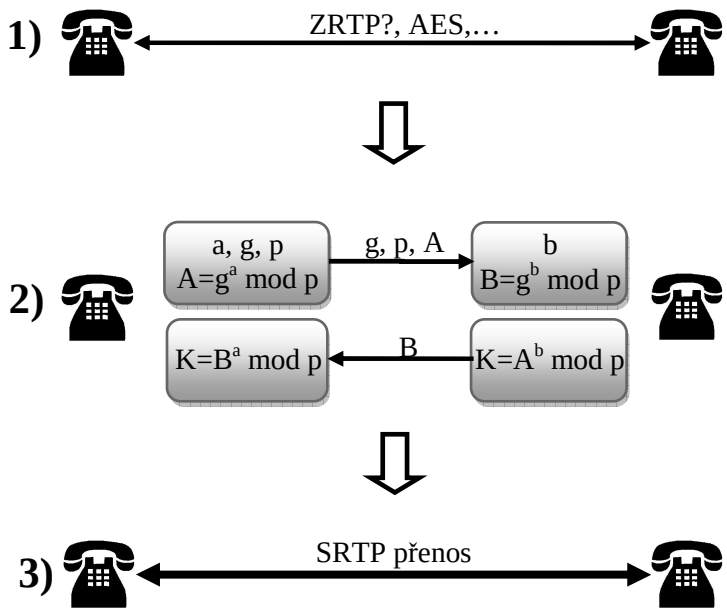
3.2.2 Funkce ZRTP

ZRTP využívá tři fáze k zajištění bezpečného přenosu dat. Nejdříve detekuje, jestli všichni účastníci podporují ZRTP implementaci, pokud ano, započne se výměna symetrických klíčů a dále proběhne přepnutí do SRTP módu.

V první fázi dochází k výměně informací o podobě šifrování, použitých klíčích a způsob autentizace mezi všemi účastníky. ZRTP specifikuje dva režimy s různou délkou klíče. AES Counter Mode (délka klíče 128 bitů) a AES Counter Mode (délka klíče 256 bitů).

Výměna klíčů ve druhé fázi proběhne pomocí Diffie-Hellmannova algoritmu a může být zajištěna dvěma módy. Mód o délce 3072 bitů a mód o délce 4096 bitů.

Jestliže proběhla výměna klíčů v pořádku, může se komunikace přepnout do SRTP módu. K tomuto kroku se přejde, až ZRTP vypočte všechna klíčová data, nastaví v SRTP kryptografickou souvislost a pomocí kontrolních informací ověří, zda úspěšně proběhla celá operace. Vše je ukázáno v obrázku 8.



Obrázek 8 fáze ZRTP

1) zjištění podpory ZRTP a výměna šifrovacích informací
2) Výpočet klíčů Diffie – Hellmanovým algoritmem a jejich výměna
3) Přepnutí na SRTP přenos

Philip Zimmerman navrhl freewarový program ZFONE pro šifrování hovoru, bez možnosti odposlouchávání. Ten používá ZRTP šifrování a běží na pozadí systému počítače. Může být použit jak u operačního systému Windows, tak u Linuxu. Nutností je instalace programu na oba komunikující počítače.

3.3 TLS

3.3.1 Popis TLS

TLS [26] slouží pro bezpečnou komunikaci v síti TCP/IP. Využívá se u poštovních služeb, datových přenosů, internetový fax, VoIP telefonie. Protokol vychází ze staršího SSL (secure socket layer), který byl po letech vývoje příliš obsáhlý, a byla snaha protokol zjednodušit. Poslední verze SSL byla 3.0, ze kterého vychází TLS 1.0. Nejnovější verze protokolu TLS je 1.2.

Pokud uživatel používá komunikaci zabezpečenou pomocí TLS, může si být vždy jistý, s kým komunikuje, a má jistotu, že spojení není odposloucháváno „třetí osobou“. To zajišťuje účinná kryptografie a autentizace uživatele i serveru. K tomu se používá zabezpečená komunikace pomocí klíčů a složitých výpočtů pro zašifrování komunikace. Vyjednávání bezpečnostních a přenosových parametrů vždy probíhá v režimu klient – server.

3.3.2 Požadavky TLS protokolu

- Bezpečné propojení dvou účastníků pomocí utajených šifer
- Součinnost: vývojáři by měli být schopni vyvíjet aplikace využívající TLS zabezpečení schopné měnit kryptografické parametry spojení bez znalosti zdrojového kódu aplikace druhého účastníka spojení.
- Rozšiřitelnost: Aby nebyla snaha vytvářet zcela nový bezpečnostní protokol, usiluje TLS o vytvoření rámce, který by byl postupně rozšiřován o nové nezbytné veřejné klíče a šifrovací metody.
- Efektivita: Protože je proces šifrování velice náročný na výkon procesoru, redukuje TLS počet spojení, která potřebují nový výpočet šifrovacího algoritmu.

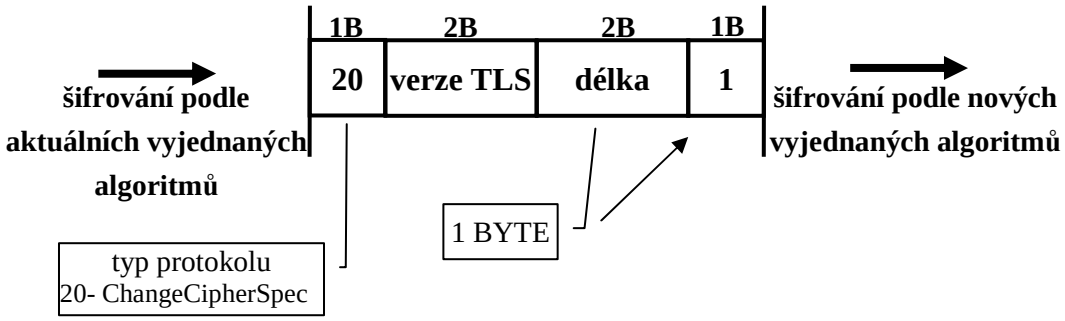
3.3.3 Rozdělení

TLS protokol lze rozdělit do dvou vrstev. Vrstvu pro přenos dat a vrstvu pro vyjednání přenosových parametrů. Toto v TLS zajišťuje několik protokolů.

1. **Record protokol -** šifrovaný přenos dat
2. **Handshaking protokol -** vyjednání přenosových parametrů
    - **Change Cipher Spec Protocol**
    - **Alert Protocol**
    - **Handshake Protocol**

3.3.4 Change Cipher Spec Protocol

Protokol pro informaci mezi účastníky, že další komunikace proběhne podle nově vyjednaného šifrování a klíče. Tato informace se posílá na konci procedury vyjednávání spojení. Informace (obrázek 9) se skládá z jedné zprávy šifrované a komprimované podle současných šifrovacích algoritmů. Zpráva má délku 1 byte. Všechny data od této chvíle budou šifrovaná podle nových algoritmů.



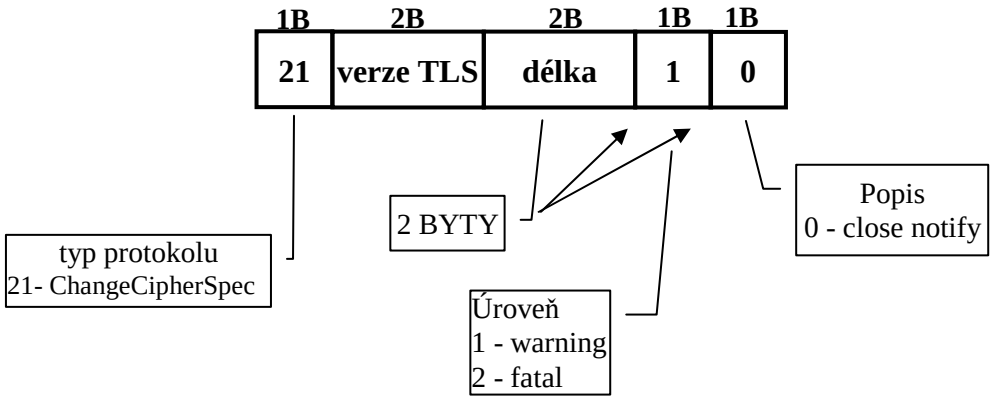
Obrázek 9 Change Cipher Spec Protocol

3.3.5 Alert protocol

Tímto protokolem probíhá informování o vzniklých chybách ve spojení. Po indikaci chyby se informace okamžitě posílá druhému účastníkovi a může nastat ukončení spojení. Výstražná zpráva (obrázek 10) obsahuje závažnost zprávy a popis výstrahy. Jestliže je výstraha na úrovni „fatal“, dojde okamžitému ukončení komunikace s protistranou.

Některé chybové výstrahy:

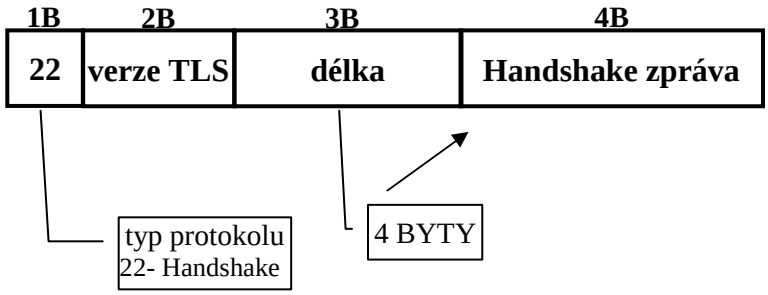
- špatně zaznamenaná MAC
- větší než povolená délka záznamu
- špatný certifikát
- odvolaný certifikát
- certifikát vypršel
- neznámý CA
- chyba dekódování
- nedostatečná bezpečnost



Obrázek 10 Alert protocol

3.3.6 Handshake protokol

Tímto protokolem (obrázek 11) si účastníci vyjednávají kompletní podobu zabezpečeného přenosu. Vyjednává se, kdy začne komunikace, verze protokolu, kódovací a dekódovací algoritmy, vzájemné ověření a použité veřejné klíče, šifrovací techniky.



Obrázek 11 Hanshake protokol

Handshake Protocol zahrnuje následující kroky:

- výměna HELLO zpráv
- výměna nezbytných kódovacích parametrů
- výměna certifikátů a šifrovacích informací
- výměna hlavního tajného kódu
- poskytnutí bezpečnostních parametrů záznamové vrstvě
- povolení ověření serveru i klientovi, že byly vypočteny stejné bezpečnostní parametry, bez zásahu útočníka

3.3.7 Hello zprávy

Používají se pro domluvu bezpečnostních parametrů pro následující spojení. Při začátku nového spojení jsou všechny předešlé parametry vynulovány a proběhne nové nastavení všech důležitých bezpečnostních prvků.

Hello regest - touto zprávou se oznamuje protější straně žádost o nové vyjednání bezpečnostních podmínek.

Client hello - Klient vysílá požadavek serveru o dohodnutí podmínek. Zpráva obsahuje nejvyšší podporovanou verzi TLS a seznam požadovaných šifrovaných metod.

Server hello - Odpověď na client hello pokud je dohodnuta shodná sada algoritmů, pokud ne, vysílá se chybové hlášení.

Server hello done - zpráva serveru klientovi s oznámením ukončení iniciační dohody na použitých algoritmech.

Certificate - zaslání certifikátu protistraně, certifikát se posílá buď jen klientovi, nebo pokud to server vyžaduje tak i jemu

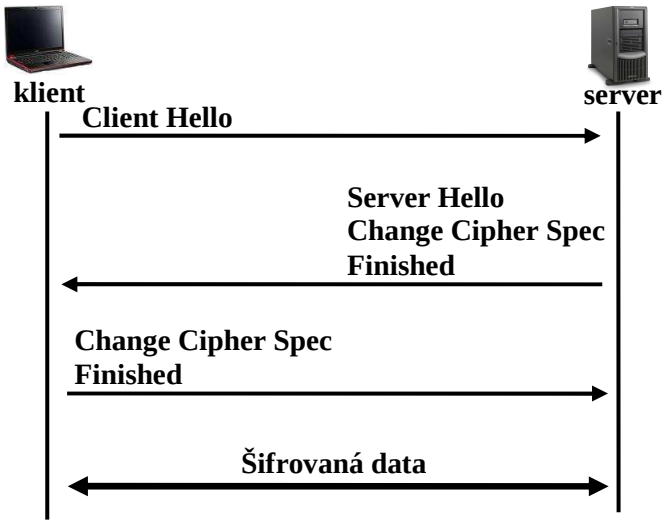
Server/client key exchange message - zpráva se zasílá hned po odeslání certifikátu serverem a to jen v případě, že certifikát neobsahuje dostatek dat pro výměnu šifrovacích tajemství

Finished - poslední zpráva oznamující, že handshake proběhl v pořádku.

Handshake neboli dohodnutí na šifrovacích parametrech může probíhat ve dvou módech. Jedná se buď o zkrácený handshake, nebo úplný handshake.

Zkrácený handshake

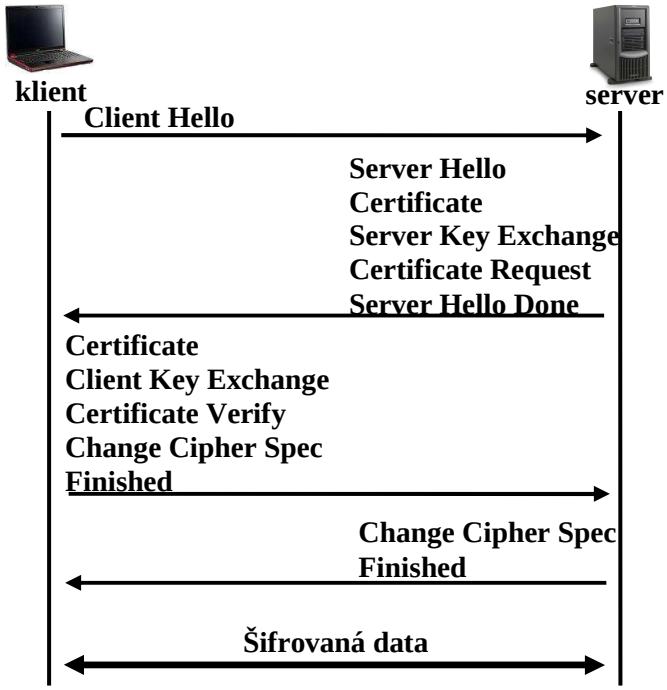
Handshake navazující na již vytvořené spojení. Probíhá pouze změna šifrovacích pravidel, ale neprobíhá autentizace na základě výměny certifikátů. Zobrazeno na obrázku 12.



Obrázek 12 zkrácený hanshake

Úplný handshake

Zde proběhne kompletní dohoda o způsobu komunikace mezi oběma stranami a proběhne autentizace pomocí certifikátů. Úplný handshake je zobrazen v obrázku 13.

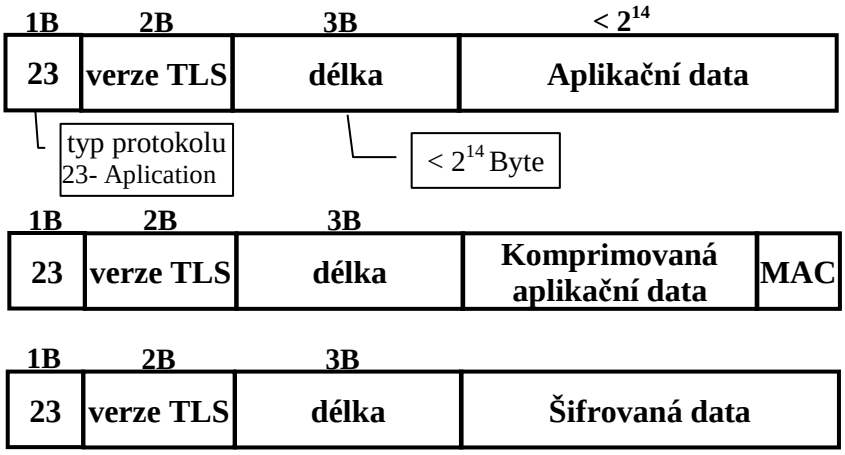


Obrázek 13 úplný hanshake

3.3.8 Record protocol

Record protokol se stará o přenos aplikačních dat, které šifruje podle předem dohodnutých šifrovacích podmínek mezi oběma uživateli, které zajistil Handshake protokol.

Je to vrstvový protokol (obrázek 14). Každá jeho vrstva obsahuje pole délka, popis a obsah. Každou zprávu určenou k přenosu rozděluje do bloků, komprimuje, kóduje a ve výsledku přenáší. Po přijetí zprávy ji rozkóduje, ověří, rozbalí, přeloží a dodá vyšší vrstvě. Jestliže přijatá zpráva není dle dohodnutého formátu, okamžitě se posílá varovná zpráva.



Obrázek 14 Vrstvy Record protocolu

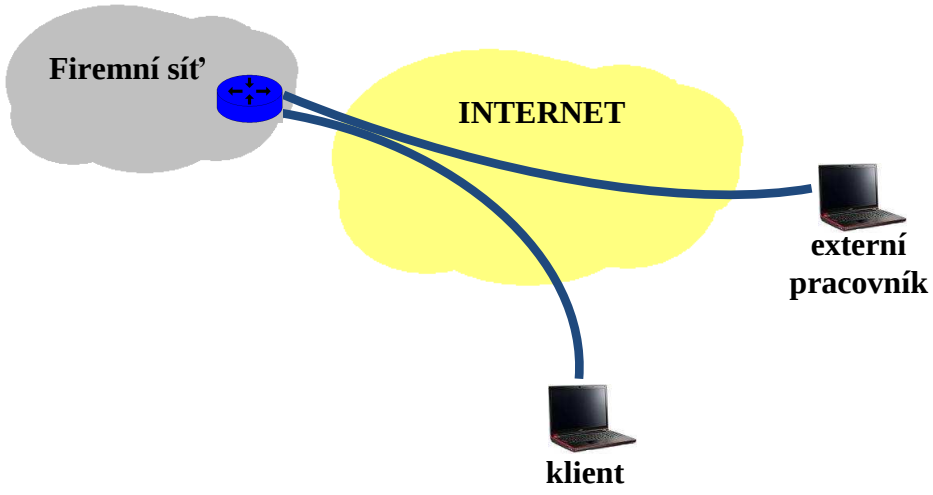
Bezpečnostní parametry Record protokolu:

- **Konec spojení** - udává zda je v tomto spojení tato entita považovaná za klienta nebo server
- **PRF algoritmus** - udává algoritmus použitý ke generování hlavního šifrovacího tajemství
- **Hromadný šifrovací algoritmus -** udává algoritmus, který je použit pro šifrování všech dat
- **MAC algoritmus** - vypočtená hodnota MAC pro ověření pravosti zprávy na základě hashe
- **Kompresní algoritmus -** použitý algoritmus pro kompresi dat
- **Master secret -** 48 bytová šifra sdílená oběmi stranami
- **client random -** 32 bytová šifra na straně klienta
- **server random -** 32 bytová šifra na straně serveru

3.4 VPN

VPN - Virtual Private Network [12, 13, 14] je virtuální síť, která zajistí bezpečné propojení dvou nebo více počítačových sítí nebo koncových bodů v síti přes nezabezpečenou síť. K propojení dvou bodů přes internetovou síť stačí mít pouze nainstalovaný příslušný software a není nutno používat jakýkoli přídavný hardware. To z VPN činí levné řešení, jak zajistit bezpečnou komunikaci přes internetovou síť. Bezpečnost přenášených dat je docílena zapouzdřováním dat, kvalitním šifrováním a používáním hashovacích funkcí. Přes

nezabezpečenou síť je vytvořen virtuální tunel, do něhož se přesměruje datový provoz. Ten poskytne datům ochranu před vnějšími útoky. Na obrázku 15 je zobrazen princip takového tunelu, kde se přes virtuální tunel firemní pracovník nebo klient připojí k firemní síti.



Obrázek 15 Princip VPN tunelu přes internetovou síť

Výhody VPN tedy činí ve snadnosti zabezpečení jakéhokoli provozu v IP síti mezi dvěma koncovými body a tím poskytnutí bezpečí před případnými útoky. Virtuální privátní sítě se ale také potýkají s řadou problémů. Patří mezi ně vysoký nárůst datového provozu mezi zabezpečenými stranami a větší zatížení při šifrování, jelikož šifrování dat a výpočet hashovacích záznamů na straně odesilatele a dešifrování u příjemce znamená velice složitý matematický výpočet, tudíž vetší zatížení procesoru. Procesem šifrování musí projít každý paket, navíc se k němu přidává vypočtený hash, takže jeho výsledná velikost po zašifrování bude větší. To klade nároky na větší přenosovou šířku pásma spojení. Problémy také můžou nastat s NAT, tedy změnou IP adres příjemce a odesilatele, v tom případě nebude souhlasit vypočtený hash a paket bude zahozen. Dalším problémem může být přenos virů přes vybudovaný virtuální tunel.

3.4.1 Módy VPN

VPN dokáže pracovat ve dvou módech.

- **Transportní mód** spojuje dva koncové body. Záhlaví zdrojové a cílové IP adresy se tedy nemění, ale veškerý obsah bude zašifrován.

- V **tunelovacím módu** je virtuální tunel nakonfigurován mezi síťové brány, datový obsah je šifrován a zdrojové a cílové IP adresy jsou změněny na IP adresy prvků, které tvoří konce tunelu.

3.4.2 Požadavky na VPN síť

Kvalitní VPN síť by měla být schopna zajistit následující požadavky:

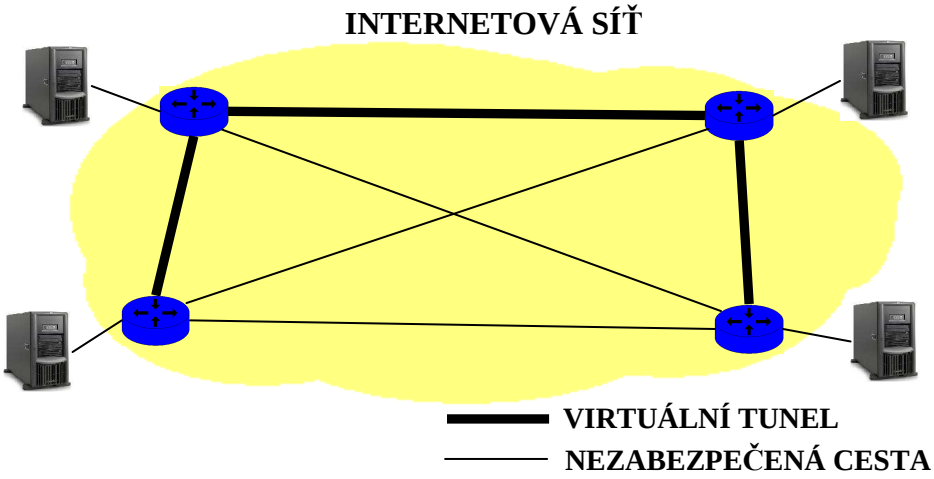
- **vytvoření bezpečného VPN tunelu** - definuje provoz, který se má zabezpečit a způsob zapouzdření a šifrování
- **autentizace obou stran tunelu** - obě strany se vzájemně autentizují pomocí předsdílených hesel a klíčů nebo bezpečněji SSL/TLS zabezpečením s uživatelskými certifikáty
- **odolnost proti odposlouchávání** - zajistí se kvalitním šifrováním přenášeného obsahu pomocí symetrických a asymetrických šifer DES, 3DES, AES, RSA
- **ochrana před změnou datového obsahu** - z každého paketu se vypočte takzvaný hash, který se přidá k paketu, příjemce obsah přečte a vypočte stejný hash, přijatý i vypočtený hash musí souhlasit, používají se hashovací funkce SHA nebo MD5
- **ochrana před znovuposíláním paketu útočníkem** - pakety se označí sekvenčními čísly, při průchodu paketu směrovačem se sekvenční číslo sníží o 1, až dosáhne nuly, paket se zahodí

3.4.3 Topologie VPN

VPN tunel lze vytvořit vždy pouze mezi dvěma body. Jestliže máme víc zařízení, mezi kterými vyžadujeme bezpečný provoz, je nutno vytvořit více tunelů. Nabízí se několik řešení jak postupovat.

Propojení všech prvků v síti - toto je nejnáročnější případ, protože je vyžadováno zabezpečené spojení mezi všemi zařízeními v síti. V tom případě musí být vytvořen tunel mezi všemi zařízeními a počet tunelů strmě roste se vzrůstajícím počtem zařízení v síti. Pro zabezpečení spojení mezi čtyřmi zařízeními je potřeba šesti tunelů. Při pěti zařízeních už potřebujeme deset tunelů.

Částečné propojení - zde se dopředu předpokládá, mezi kterými prvky sítě bude největší provoz. Virtuální tunel se tedy vytvoří jen mezi routery a servery a ostatní provoz se nezabezpečí (obrázek 16). To vyřeší problém s vysokým počtem tunelů. Chceme-li ale zabezpečit provoz mezi dvěma zařízeními, které nemají vytvořený tunel, musíme provoz přesměrovat přes bezpečnou cestu.



Obrázek 16 částečné propojení virtuálními tunely

Hvězda - zde se vytvoří virtuální tunely mezi centrálním prvkem sítě a ostatními zařízeními. Opět zde dochází k úspoře počtu vytvářených tunelů

Dnes již existuje několik druhů VPN technologii, z nichž se každá snaží zajistit zabezpečení dat odlišným způsobem. Některé druhy VPN technologií - IPsec (IP security), GRE (Generic Routing Encapsulation), PPTP (Point-to-Point Tunneling Protocol), L2TP (Layer 2 Tunneling Protocol).

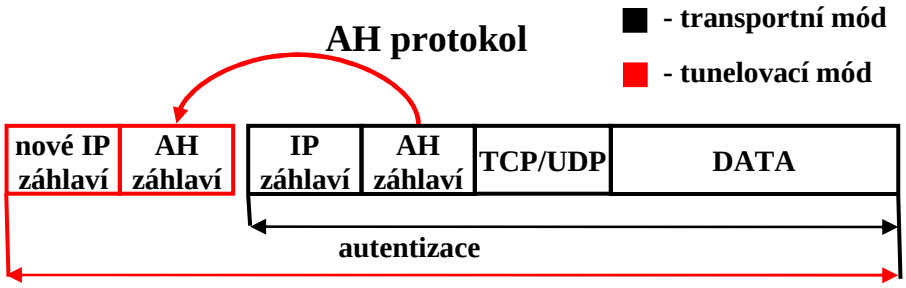
3.5 Ipsec

Ipsec (Internet Protocol Security) [12, 17, 18, 19, 30] slouží pro zabezpečení datového provozu v nechráněné síti. Nejčastěji bývá implementován na routery, které tvoří bránu mezi lokální sítí a například internetovou sítí. Ipsec je zabezpečení typu end-to-end, což nám dává možnosti zabezpečení například dvou firemních sítí propojených přes internetovou síť, nebo například externího pracovníka, který se chce bezpečně připojit k firemnímu serveru kdekoli z internetu.

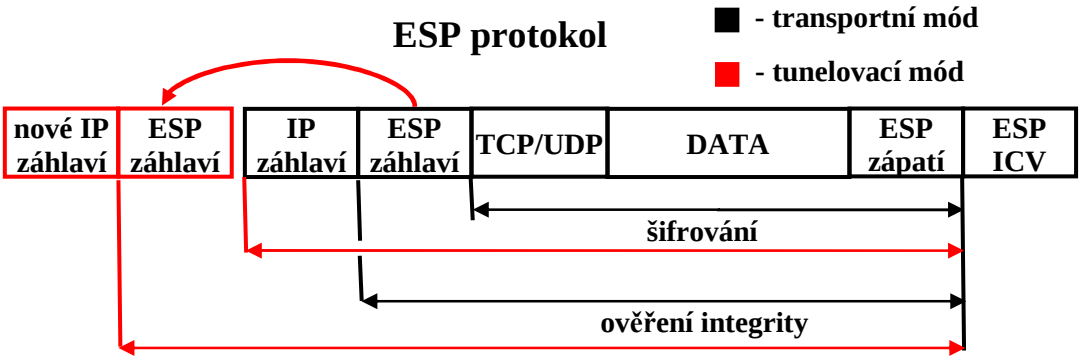
Poskytuje ochranu proti odposlouchávání, falšování identity, změně obsahu paketů. K tomu využívá soustavu, bezpečnostních protokolů, kryptografické algoritmy a šifrovací klíče (IKE, Diffie - Hellman, AES, DES, HMAC, SSL/TLS, atd.) Bezpečná asociace se může skládat z autentizace obou protistran na základě SSL/TLS s využitím certifikátů X.509 podepsaných certifikační autoritou, zašifrování paketů dohodnutou AES šifrou a ověření obsahu HMAC hashovaní funkcí. Každý přenášený paket je označen 32 bitovým SPI kódem podle něhož

příjemce rozezná jakou bezpečnostní asociací byl paket zabezpečen. Ipsec pracuje na třetí vrstvě OSI modelu, proto poskytuje ochranu kterékoli síťové aplikaci.

K přenosu dat je možno zvolit dva ipsec módy. **Transportní mód** se používá pro bezpečný provoz mezi dvěma koncovými počítači. K tomu je využita autentizace obou stran a podle použitého protokolu taky šifrování obsahu. **Tunelovací mód** zastává funkci bezpečné brány. Zabezpečuje komunikaci mezi dvěma sítěmi propojenými třetí nezabezpečenou sítí. Tato varianta se implementuje do routerů tvořící gateway sítě. Opět se obě strany autentizují, přidá se nové IP záhlaví a šifruje se obsah. Oba dva módy mohou využívat dva protokoly, ESP a AH (obrázky 17 a 18). ESP protokol autentizuje protistrany, šifruje obsah a ověří integritu, AH pouze provede autentizaci a integritu.



Obrázek 17 zapouzdření AH a ESP protokolem

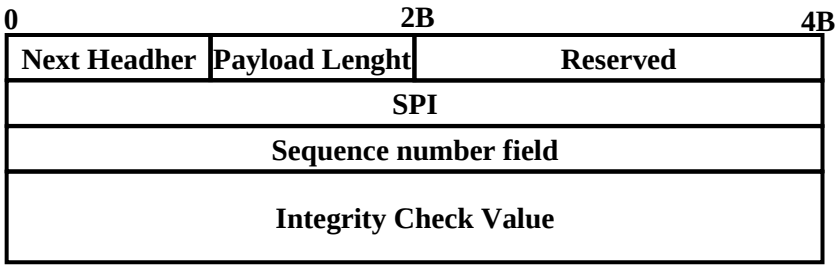


Obrázek 18 zapouzdření ESP protokolem

3.5.1 AH protokol

AH (Authentication Header) [31] zajišťuje autentizaci a integritu přenášených pomocí hashovací funkce HMAC. Dále ochrání pakety proti znovuposílání sekvenčními čísli, kterými je označen každý paket. V transportním módu vkládá AH záhlaví (obrázek 19) mezi IP záhlaví a

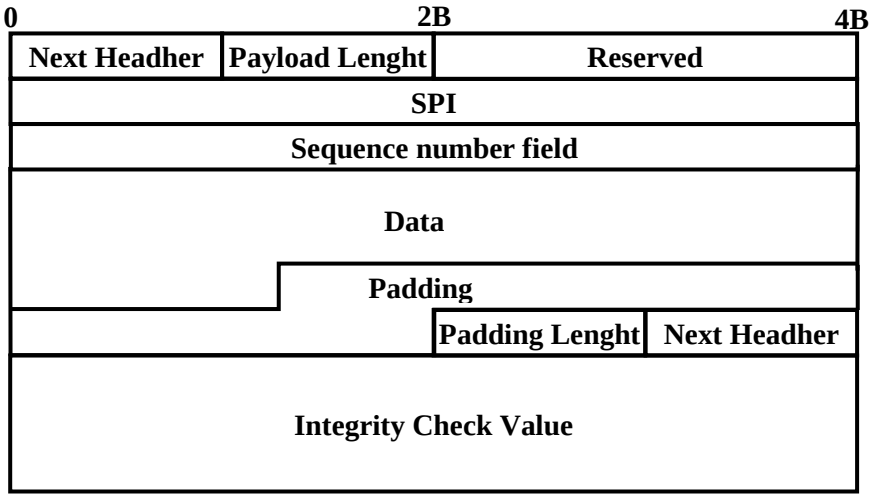
záhlaví dalšího protokolu (TCP, UDP, …) nebo dalších IPsec záhlaví, protože paket může po své cestě k cíli projít několika zabezpečenými trasami. Tunelovací mód přidá nové IP záhlaví nesoucí adresu konce tunelu, ale ne cílovou IP adresu. IP adresa cíle je uložena v záhlaví za AH záhlavím. V tomto módu je možný přenos IPv4 přes Ipv6 a naopak. Původní IP záhlaví je chráněné, tudíž například IPv4 může projít přes IPv6 tunel.



Obrázek 19 Formát AH záhlaví

3.5.2 ESP protokol

ESP (Encapsulating Security Protocol) [32] stejně jako AH protokol zajistí autentizaci, integritu a ochranu před znovuposíláním, ale navíc poskytne důvěrnost pomocí šifrování šiframi AES, DES, 3DES. V transportním módu je ESP záhlaví vloženo stejně jako AH mezi IP záhlaví a záhlaví dalšího protokolu. Za zašifrovanou datovou částí je vloženo ESP zápatí a ICV pole (Integrity Check Value), sem se vloží vypočtený hash. Struktura ESP paketu je zobrazena na obrázku 20. V tunelovacím režimu je přidáno nové IP záhlaví konce tunelu, za něj ESP záhlaví. Všechny následující pole (staré IP záhlaví, datová část a ESP zápatí) jsou šifrovány. Z šifrované části se vypočte hash a vloží se do pole ICV.



Obrázek 20 Struktura paketu šifrovaného pomocí ESP

3.5.3 IPsec SA

IPsec SA - IPsec Security Asociantion je bezpečnostní asociace, na které se dohodnou oba konce tunelu. Pro každý směr datového provozu se vyjedná vlastní IPsec SA. Vyjednává se řada bezpečnostních parametrů jako je mód činnosti, protokol pro zapouzdření paketů, konce tunelu, zabezpečovaný provoz, doba trvání a SPI index (Security Parameter Index). SPI je přiřazen konkrétnímu spojení podle koncových IP adres, portu a protokolu. Podle SPI pozná příjemce jak zprávu dešifrovat.

3.5.4 ISAKMP

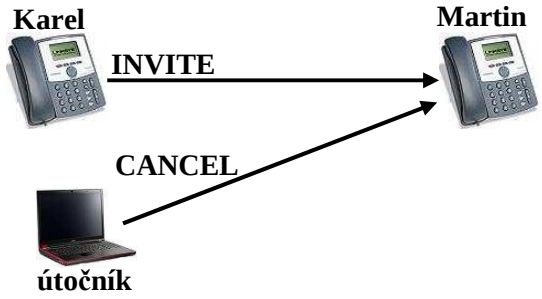
ISAKMP - Internet Security Association and Key Management Protocol slouží pro správu IPsec SA, autentizaci obou konců tunelu, vytváření, údržbě a ukončení IPsec SA a vytváření a výměna dynamických klíčů. ISAKMP je obecný protokol, není navržen pouze pro IPsec, ale lze jej použít pro výměnu šifrovacích informací také pro jiné protokoly. Obsah paketu nemá jednotnou podobu. Konkrétní informace jsou obsaženy v části payloads a jejich obsah a podobu určí odesílatel v závislosti na bezpečnostních podmínkách, které chce vyjednat. Pro komunikaci používá ISAKMP UDP port 500.

3.5.5 IKE

IKE - Internet Key Exchange je protokol, který se používá při vytváření IPsec SA a při výměně bezpečnostních klíčů. Pracuje ve dvou fázích. První fáze IKE je proces vytváření IKE SA - bezpečného spojení, které bude využito v druhé fázi. První fáze používá dva módy - hlavní a agresivní. Komunikace hlavním módem proběhne šesti zprávami, agresivním pouze čtyřmi. Agresivní mód je tedy rychlejší, ale méně bezpečný. Ve druhé fázi IKE se vytvoří pomocí D-H algoritmu sdílený klíč pro šifrování datového provozu a vytvoří se dvě protisměrné IPsec SA. Komunikace probíhá jenom jedním módem - rychlý mód.

4 Metody autentizace v protokolu SIP

Nejsložitější částí IP telefonie je přenos signalizace při hovoru, při komunikaci klienta se serverem, nebo vzájemná komunikace mezi servery. Při komunikaci je vždy nutno znát, s kým daná strana komunikuje. SIP nabízí několik druhů autentizace komunikujících stran lišící se v míře bezpečnosti ověření. Ve VOIP telefonii je ověření všech klientů důležitým krokem, který kromě základního účelu poskytnout služby VOIP serveru pouze povoleným účastníkům, plní ochranou funkci před útoky využívající rušení hovorů CANCEL zprávami (obrázek 21), zahlcování serveru registračními požadavky a podobné typy útoků.



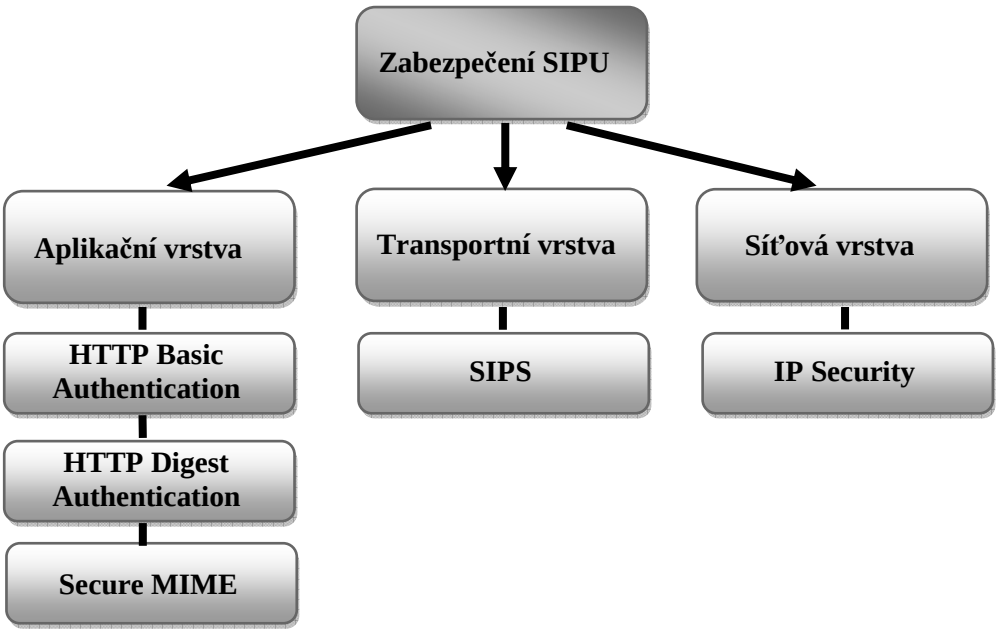
Obrázek 21 rušení žádosti o spojení útočníkem

Autentizace je proces ověření, zda žádost uživatele je legitimní a jestli pochází od klienta, který má povolení využívat danou službu. Autentizace v SIPu může proběhnout mezi UA (user agent) a serverem (proxy, registrar, UAS), kde server požaduje po uživateli ověření totožnosti před zpracováním žádosti. Stejně tak může UA požádat o ověření serveru - vzájemná autentizace.

Autentizace je vyžadovaná v těchto případech:

- **Registrace k SIP serveru** - zabránění využití služeb nepovolaným účastníkům a útočníkům
- **Žádost o službu** - například v žádosti o zprostředkování hovoru zprávou INVITE se volající identifikuje volanému
- **Změna relace** - ochrana před změnou relace třetí osobou (útočníkem), například přidání se útočníkem do konferenčního hovoru
- **Ukončení relace** - ochrana před ukončením hovoru útočníkem

V protokolu SIP je několik bezpečnostních opatření, jak zajistit autentizaci účastníků (obrázek 22). Autentizaci je možno povolit na různých vrstvách. Na aplikační vrstvě ji zajišťuje HTTP Authentication a S/MIME, na transportní vrstvě je to SIPS a na síťové vrstvě lze zajistit autentizaci přes IPsec.



Obrázek 22 Autentizace SIP protokolu na různých vrstvách

4.1 HTTP Basic Authentication

Tento typ autentizace vychází z HTTP komunikace. Je používán pro přístup k stránkám chráněným heslem z internetových prohlížečů. Je to základní typ autentizace pomocí sdíleného hesla, které je při přenosu chráněno kódováním BASE64. Algoritmus Base64 převádí znaky hesla po trojicích do ASCII hodnot, z jejich binární podoby vytvoří 4 indexy, které převede do formy písmen a čísel a znaků. Obsah zprávy zůstane nešifrován. Http Basic authentication [29] poskytuje nejmenší míru bezpečnosti. V současné době se již v SIPu nepoužívá a byla nahrazena Digest autentizací.

4.2 HTTP Digest Authentication

V SIPu základní autentizační metoda. Vychází z Basic autentizace, kde se také používá sdílené heslo, ale v tomto případě se neposílá přímo, ale ve vypočteném hashi. Při žádosti o registraci nejdříve pošle klient žádost REGISTER. Tato zpráva ale ještě neobsahuje registrační data, jsou vyplněna zatím jenom pole Via, Contact, To a From. Tato žádost bude serverem ihned odmítnuta v odpovědi 401 Unauthorized. Zároveň bude klient v této zprávě požádán o zaslání registračních údajů pomocí Digest Authentication. Tyto údaje musí zaslat zašifrované pomocí algoritmu MD5. K tomuto účelu mu server vygeneroval náhodné číslo *„nonce value“*.

WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="18eebd32"

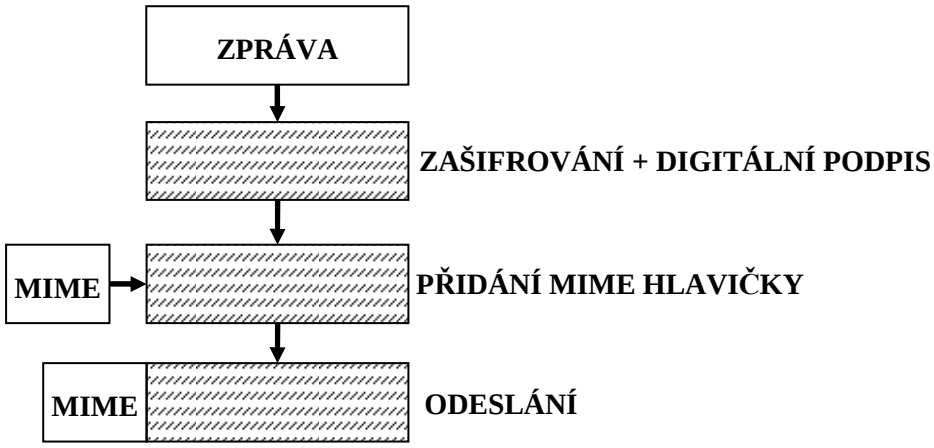
Klient vypočte hash ze svého přihlašovacího jména, hesla, z obsahu pole *realm* a *nonce*. Po té znovu pošle zprávu REGISTER, která již obsahuje zmiňované přihlašovací údaje.

Authorization: Digest username="lojza",realm="asterisk",nonce="18eebd32",uri="sip: 128.146.10.103",response="3bf8e87a0e3d493091eab03c8ed39f31",algorithm=MD5

Server přijatou hash zprávu uloženou v poli *response* porovná s vlastním vypočteným hashem a pokud se shodují, klient bude přihlášen. Tato metoda přináší větší bezpečnost při autentizaci, jelikož ze zachycení takové zprávy útočníkem nelze zjistit přihlašovací heslo klienta. Nicméně ani tento typ autentizace nijak nechrání celou právu.

4.3 S/MIME

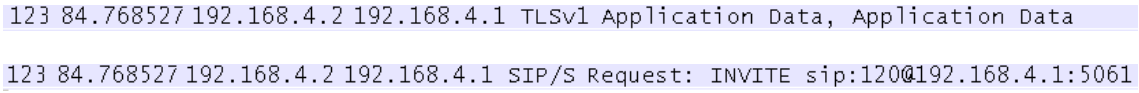
Secure Multipurpouse Internet Mail Extension [11] je formát primárně určen pro poskytnutí bezpečí v emailové poště. Definuje podobu těla zprávy, aby ji bylo možné přečíst různými poštovními servery. Díky tomu že SIP vychází z HTTP, lze tuto metodu přenést také na SIP zprávy. Obsah zprávy bude zašifrován a bude poskytnuta integrita zprávy. Proces je zobrazen v obrázku 23. Pro tvorbu bezpečné zprávy využívá S/MIME normu PKCS-7. Touto metodou budou data digitálně podepsaná a zašifrovaná. Zpráva ve formátu *application/pkcs7-mime* je ovšem zašifrovaná a běžným klientem nečitelná, pokud nepoužije nástroj k dešifrování. S/MIME nabízí tedy druhou možnost, kdy se pomocí *multipart/signed* zpráva rozdělí na dvě části, nešifrovaný text a šifrovaný podpis podle PKCS-7. Klienti se autentizují uživatelskými certifikáty X.509, což přináší maximální míru důvěry. Šifrování obsahu zprávy je zajištěno symetrickými šiframi AES, DES a 3DES.



Obrázek 23 Proces zabezpečení zprávy pomocí S/MIME

4.4 SIPS

Secure SIP je bezpečnostní mechanismus vytvořený přímo pro SIP protokol na rozdíl od S/MIME, který je převzatý z HTTP. Autentizace je v tomto případě řešena přes TLS zabezpečení a jsou použity opět klientské certifikáty X.509. Jedná se o kompletní šifrování SIP signalizace s použitím DES, 3DES a AES šifer. Pro komunikaci se používá port 5061 a protokol TCP. Nutností u SIPS je realizovat TLS zabezpečení mezi každými dvěma prvky v síti. V hlavičce zprávy bude navíc **transport=TLS** a změní se číslo portu na **5061** (obrázek 24).



Obrázek 24 Šifrovaná a dešifrovaná INVITE zpráva

INVITE sip:120@192.168.4.1:5061 SIP/2.0

Via: SIP/2.0/TLS 192.168.4.2:5061;branch=z9hG4bK76405a0e;rport

Max-Forwards: 70

From: "lojza" <sip:2220@192.168.4.2>;tag=as34b516c8

To: <sip:120@192.168.4.1:5061>

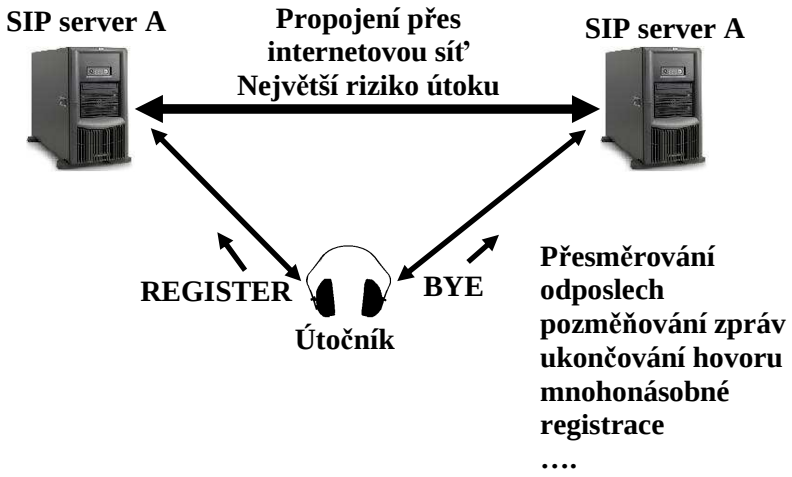
Contact: <sip:2220@192.168.4.2;transport=TLS>

Call-ID: 28e5a3737df4ca40298e1cab4f14808d@192.168.4.2

CSeq: 103 INVITE

5 Realizace zabezpečení SIP serverů

Zabezpečení VoIP ústředen je velice důležité, jestliže je spojen hovor dvou účastníků přes nezabezpečenou síť. Typickým příkladem je propojení dvou firemních sítí přes internet. Správci těchto sítí je mohou vnitřně zabezpečit, ale jakmile jde hovor internetem, je vysoké riziko napadení, protože samotná internetová síť zabezpečena není. Právě zde je riziko napadení největší a je jisté, že žádná prosperující společnost nechce přijít o svá důvěrná data. Proto je nutno u takto propojených serverů zajistit bezpečnost jak hovorových dat, tak signalizačních zpráv. Zabezpečení signalizace je důležité, protože na tuto část spojení existuje nespočet útoků. Příkladem může být útok MITM (man in the middle), který pozměňováním signalizačních zpráv přesměruje hovor přes sebe a může ho měnit, nebo jen odposlouchávat důvěrná data (obrázek 25). Dalším nepříjemným útokem na signalizaci může být útok využívající ukončování spojení pomocí posílání BYE zpráv, nebo útok na server, který zahltí tisíci registračními požadavky, čímž celý server zahltí a postaví mimo provoz.



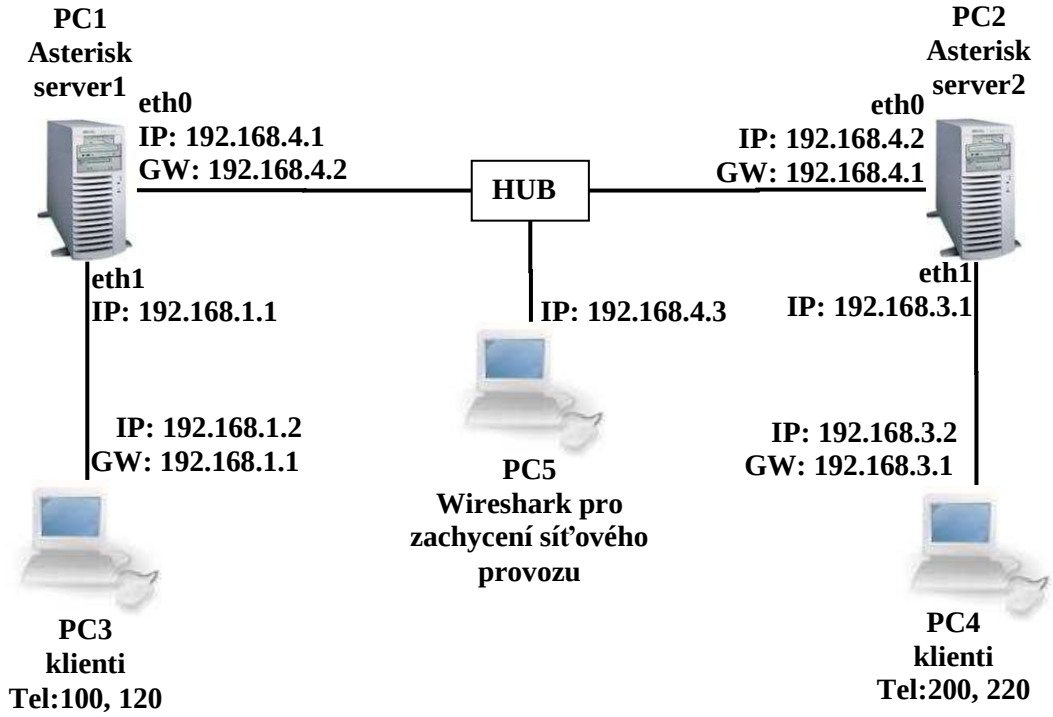
Obrázek 25 Napadení VoIP hovoru

Pro realizaci zabezpečení SIP serverů jsem použil několik ověřených zabezpečovacích implementací. Zejména byl kladen důraz na použití bezpečnostní technologie SSL/TLS, která zajistí téměř neproniknutelnou bariéru mezi důvěrným obsahem přenášených dat a útočníkem, který touží po získání těchto dat. SSL/TLS zajistí autentizaci komunikujících stran a šifrovaný přenos dat využívající pokročilých šifrovacích mechanismů. Jako SIP servery jsem použil hojně využívanou VoIP ústřednu Asterisk. Ten tvoří kompletní softwarovou telefonní ústřednu se všemi funkcemi, které jsou dnes vyžadovány od kvalitní ústředny. Jeho zabezpečení bylo provedeno na bázi VPN a IPSEC zabezpečení. Obě implementace se liší způsobem zajištění bezpečnosti dat, avšak v konečné fázi jsou obě řešení téměř neprolomitelné. Poskytují několik

úrovní zabezpečení, v nejvyšší úrovni obě shodně využívají SSL/TLS. Je důležité se zmínit, že VPN technologie ani IPSEC nebyly navrženy pro poskytnutí bezpečnosti voip technologii, ale jejich úkolem je zabezpečit celý IP protokol, ať už ho pro přenos dat využívá jakákoli technologie. Konkrétní VPN implementaci jsem použil OpenVPN a pro IPSEC jsem použil OpenSWAN. Jako alternativu k těmto řešením jsem použil zabezpečení signalizačních zpráv přes SIPS a šifrování hovorových dat pomocí SRTP. SIPS opět využívá TLS s uživatelskými certifikáty a k šifrování SRTP je použita AES šifra.

5.1 Topologie sítě

- Celá konfigurace byla prováděna na čtyřech fyzických počítačích plus jeden na zachycení síťového provozu mezi servery. Zobrazeno v obrázku 26.
- Na PC1 a PC2 byl nainstalován operační systém Ubuntu 9.10 a jako SIP server byl použit Asterisk 1.6.2, který je již obsažen v systému
- Na PC1 a PC2 byly nainstalovány aplikace OpenVPN a OpenSWAN
- Na PC3 a PC4 byl nainstalován operační systém Windows XP
- Na obou klientských počítačích byli nainstalovaní softwaroví voip klienti SJ Phone a X-lite
- servery byly propojeny sítí 192.168.4.0 a každý pro své voip klienty využíval podsíť. 192.168.1.0 a 192.168.3.0



Obrázek 26 topologie sítě

5.2 Základní nastavení počítačů

Počítačům jsem přiřadil pevnou IP adresu a bylo nutno nastavit gateway pro komunikaci mimo vlastní síť. Počítačům, které využívaly dvě síťová rozhranní bylo nutno nastavit ip forwarding pro předávání paketů mezi jednotlivými rozhranními. Toto minimum na úrovni síťování postačí pro pozdější plně funkční chod všech dalších implementací.

PC1: ***ifconfig eth0 192.168.4.1/24***

ifconfig eth1 192.168.1.1/24

route add default gw 192.168.4.2 - nastavení gateway

echo 1 > /proc/sys/net/ipv4/ip_forward – nastavení předávání paketů z jednoho síťového rozhranní na druhé

PC2 obdobné nastavení, IP adresy - 192.168.4.2 (eth0), 192.168.3.1 (eth1), gateway - 192.168.4.1

PC3: **Adresa IP: 192.168.1.2**

Maska podsítě 255.255.255.0

Výchozí brána: 192.168.1.1

PC4 - IP 102.168.3.2 a brána 192.168.3.1

5.3 Asterisk

Asterisk plní funkci kvalitní open - source softwarové telefonní ústředny bežící na operačním systému Linux a Unix. Je to ideální PBX především díky široké podpoře řady Voip protokolů (SIP, H.323, MGCP, VoFR) a protokolů pro realizaci klasické telefonní sítě (POTS, ISDN). Já jsem použil verzi Asterisku 1.6.2 s podporou TLS. Tuto verzi obsahuje mnou používaný operační systém Ubuntu 9.10, tudíž není nutná kompilace ze zdrojového kódu. Instalace je tedy provede příkazem ***apt-get install asterisk***. Po nainstalování přejdeme k základnímu nastavení Asterisku provedenou v souborech **sip.conf**, **iax.conf** a **extensions.conf**.

5.3.1 Konfigurace Asterisku

Soubor **sip.conf** je rozdělen do několika sekcí. Základ tvoří sekce **[general]**, ve které se definují některé obecné nastavení platné pro všechny klienty, používající SIP protokol. Zpravidla se zde uvádí **context** pro přiřazení do skupiny příchozích volání, povolení a zakázání vybraných kodeků, port, nastavení NAT a další. Tuto sekci je možno nechat prázdnou za předpokladu, že konfigurace každého klienta bude obsahovat vlastní nastavení všech důležitých parametrů. Další sekce již obsahuje nastavení klientů.

[general] - v sekci general se nastaví chování pro všechny klienty společně

context=server1 - odkaz na číslovací plán v extensions.conf

disallow=all - zakázaní všech audio kodeků

allow=alaw - povolení kodeku G.711 s logaritmickou kompresí A-law

[zbynek] - nastavení klienta zbynek

type=friend - povolení odchozích i příchozích hovorů

callerid=zbynek<100> - identifikace volajícího

secret=zbynek1 - heslo

host=dynamic - ip adresa se přidělí na základě registrace

Soubor **extensions.conf** slouží jako dialplan. Zde se nastavuje chování příchozích a odchozích hovorů. Opět je zde možno využít sekce [general], další sekce slouží pro ovládání skupin volání. Konfigurace se skládá z jednotlivých pravidel, které mají následující podobu -

exten=>name,priority,application

Jako ***name*** se obvykle volí telefonní číslo, ***priority*** označuje prioritu pravidla a ***application*** značí příkaz, který bude proveden.

exten=>100,1,Dial(SIP/zbynek) -volané číslo **100** bude směrováno přes SIP uživateli **zbynek**

V asterisk serveru 1 jsem nakonfiguroval klienty **zbynek** <100> a **radek** <120>. U druhého serveru klienty **josef** <200> a **lojza** <220>. Konfigurace proběhla obdobně jako u serveru 1. Toto je nastavení pro realizaci hovorů mezi klienty pod jedním serverem. V dalším kroku je potřeba nastavit propojení obou serverů.

5.3.2 Asterisk Trunk

Propojení obou asterisk serverů je možno nastavit pomocí SIP protokolu, kde bude komunikace probíhat stejnými SIP zprávami jako při komunikaci SIP klientů se serverem, nebo pomocí IAX (Inter Asterisk Exchange) protokolu, který byl vyvinut samotnými vývojáři Asterisku.

V sip.conf se trunk nastaví jako další klient, pouze mu nebude přiřazeno telefonní číslo a musí se nadefinovat IP adresa serveru, ke kterému se bude připojovat.

[trunk1]

type=peer - pouze pro odchozí hovory

host=192.168.4.2 - IP adresa serveru 2

username=trunk2 - identifikace

secret=heslo2 - heslo

Podoba souboru iax.conf bude stejná jako nastavení trunku v sip.conf

Dále je nutno v souboru **extensions.conf** nastavit chování trunku. Přidají se následující řádky:

exten=>_2.,1,Set(CALLERID(num)=2${CALLERID(num)})

exten=>_2.,2,Dial(SIP/trunk1/${EXTEN:1})

Z uvedeného nastavení jde vidět, že při volání se serveru 1 klientovi na server 2 zadáme před číslo volaného dvojku a naopak pro volání na server 1 zadáme před volané číslo jedničku. Pro propojení serverů se použije SIP protokol. Volba použitého kanálu pro trunk se mění úpravou obsahu závorky za příkazem Dial. Pokud chceme zajistit propojení IAXem, přepíšeme SIP na IAX2.

exten=>_1.,2,Dial(SIP/trunk2/${EXTEN:1})

exten=>_1.,2,Dial(IAX2/trunk2/${EXTEN:1})

5.4 OpenVPN

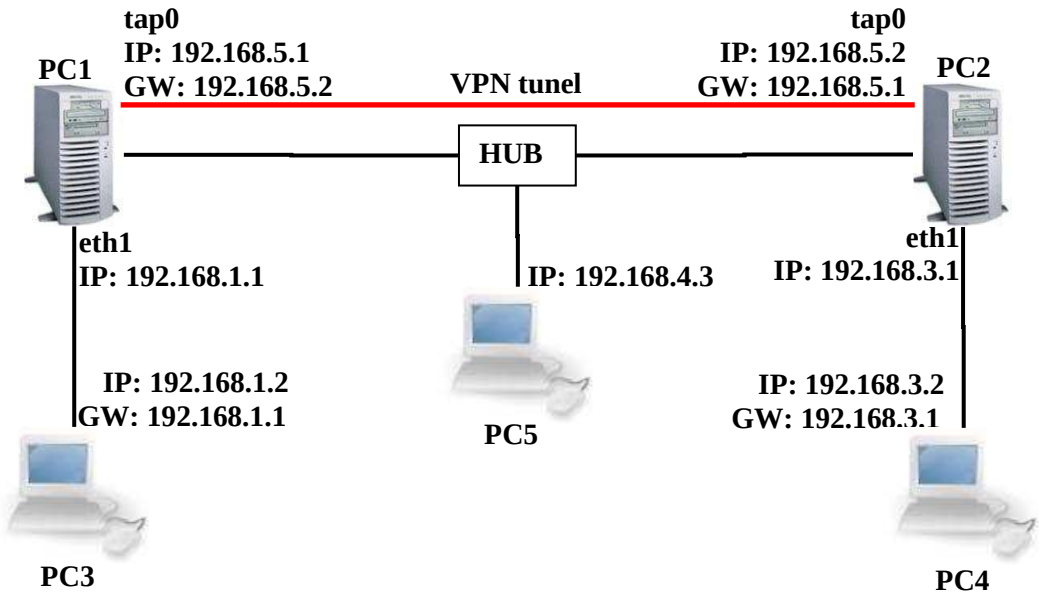
Technologie Virtual Private Network poskytuje vysokou míru zabezpečení přenášených dat, především díky využití pokročilých bezpečnostních mechanismů, mezi něž patří podpora SSL/TLS zabezpečení, poskytování důvěrnosti vyspělými šifrovacími metodami a integrita dat na základě ověřování pomocí hashovacích funkcí. Data po své cestě k cíli prochází vytvořenou virtuální sítí, která jim poskytne vysokou míru zabezpečení. Pokud budeme takto zabezpečeným tunelem provozovat VoIP telefoni a útočník datový provoz ve virtuálním tunelu zachytí, tak ze zachycených paketů rozezná pouze IP adresy začátku a konce tunelu, což nevypovídá nic o skutečném odesilateli ani příjemci, protože tunel může být nakonfigurovaný mezi routery, které tvoří hraniční brány do internetu, nerozezná ani že jde o přenos RTP hlasových paketů ani jejich datový obsah, protože data jsou kvalitně šifrovaná. Z tohoto pohledu se OpenVPN jeví jako ideální nástroj pro zabezpečení IP telefonie.

Konkrétně zabezpečení pomocí VPN řeší projekt OpenVPN. Je volně ke stažení na stránkách vývojového týmu a podporuje ho řada linuxových distribucí. OpenVPN je také možné spustit ve Windows, Mac OS X, Solaris atd. Pro vytvoření tunelu využívá virtuálních rozhraní TUN a TAP. V konfiguraci je možno zvolit, které rozhranní bude použito. Pro šifrování a autentizaci využívá OpenSSL knihovny.

Instalace OpenVPN v Ubuntu probíhá pouze příkazem ***apt-get install openvpn*** a není třeba nic kompilovat ani doinstalovat žádné doplňující balíčky. OpenVPN se konfiguruje v

souborech s koncovkou .conf, kde je nakonfigurován vpn tunel a soubor secret.key, kde je umístěn sdílený klíč v případě použití autentizace pomocí sdíleného klíče. Tyto soubory jsou umístěny v adresáři /etc/openvpn/. Po startu OpenVPN se tento adresář prohlédne a v případě nalezení těchto souborů dojde vytvoření tunelu.

Autentizace obou konců tunelu lze nakonfigurovat dvěma způsoby. První méně bezpečný způsob, je autentizace pomocí sdíleného klíče, který si uživatel předem vygeneruje a bezpečně přenese na druhou stranu spojení. Další a z pohledu bezpečnosti nejlepší volbou je autentizace pomocí TLS s x.509 certifikáty. Pro tuto volbu je nutno nejdříve vygenerovat certifikáty a podepsat je certifikační autoritou. Po nakonfigurování se vytvoří nové síťové rozhranní tap0 nebo tun0. VPN tunel je zobrazen na obrázku 27.



Obrázek 27 Vytvoření virtuálního tunelu

5.4.1 Konfigurace OpenVPN se sdíleným klíčem

V předchozí kapitole jsem se zmínil, že tato varianta je méně bezpečná než varianta s TLS. To je sice pravda, ale nelze říci, že je riziko prolomení šifry vysoké. Sdílený klíč má délku 2048 bitů, což zaručuje vysokou bezpečnost, problém ale je, předání klíče druhé straně, jestliže se útočník tohoto klíče zmocní, už mu nic nebrání k napadení spojení.

Nejdříve je nutno vygenerovat sdílený RSA klíč. To se provede pomocí příkazu:

openvpn --genkey --secret /etc/openvpn/secret.key

Správnost klíče můžeme ověřit příkazem:

openvpn --test -crypto --secret /etc/openvpn/secret.key -následně bude vytvořeno několik paketů, na kterých bude vyzkoušeno zašifrování a dešifrování sdíleným klíčem.

Hlavní konfigurace se provede v souboru *.conf. Zde jsou nejdůležitější parametry **remote, ifconfig a secret**, kde se nastaví IP adresa protistrany, IP adresa tunelu a cesta k sdílenému klíči.

remote 192.168.4.2 -IP adresa protějšího počítače

ifconfig 192.168.5.1 255.255.255.0 - IP adresa tunelu

secret /etc/openvpn/secret.key - umístění sdíleného klíče

Konfigurační soubory na obou stranách mají stejnou podobu, změna je jen v IP adresách u parametrů **remote** a **ifconfig**.

Spuštění OpenVPN se provede příkazem:

/etc/init.d/openvpn/start

V tuto chvíli máme aktivní virtuální tunel a už nám zbývá jen úprava v asterisku a úprava routingu.

5.4.2 Úprava nastavení Asterisku v sip.conf

U asterisku v **sip.conf** je nutno změnit IP adresy v nastevení trunku v řádku **host** na nové IP adresy rozhranní tap0.

sip.conf (asterisk1)

host=192.168.5.2

sip.conf (asterisk2)

host=192.168.5.1

5.4.3 Nastavení routingu

Posledním krokem je přepsání routovací tabulky v PC1 a PC2, kde se změní gateway, aby datové pakety při výstupu ze sítě použily vytvořený virtuální tunel.

PC1: ***del default gw 192.168.4.2*** **(192.168.4.1 PC2)**

add default gw 192.168.5.2 **(192.168.5.1 PC2)**

5.4.4 Generování TLS certifikátů

Pro OpenVPN s autentizací pomocí X.509 certifikátů je nutné je nejdříve vygenerovat a podepsat certifikační autoritou. K vygenerování všech certifikátů jsou použity ssl knihovny,

které již operační systém obsahuje. Bude vygenerovaná certifikační autorita, server certifikát, klientský certifikát a soubor s diffieho-hellman parametry pro výměnu klíčů.

Nejdříve zkopírujeme skripty pro tvorbu certifikátů z adresáře

/usr/share/doc/openvpn/examples/easy-rsa/2.0 do adresáře

/etc/openvpn/easy-rsa/

V souboru **vars** upravíme délku šifrovacího klíče (**export KEY_SIZE=2048**), dobu platnosti certifikační autority (**export CA_EXPIRE=3650**) a uživatelských certifikátů (**export KEY_EXPIRE=3650**) a další identifikační údaje (COUNTRY, PROVINCE, …) pro generování certifikátů.

Nyní už můžeme generovat certifikáty. Přejdeme do adresáře /etc/openvpn/easy-rsa, který obsahuje potřebné skripty.

- **. ./vars** -aktivace proměnných
- **./clean-all** -vymazání předchozích certifikátů
- **./build-ca** -generování certifikační autority, vyplníme několik identifikačních údajů, vytvoří se soubory **ca.crt** a **ca.key**
- **./build-server server** -generování server certifikátu a jeho klíče a podepsání certifikační autoritou, vygeneruje se soukromý RSA klíč a opět jsme vyzváni k vyplnění několika údajů a hesla, vytvoří se soubory **server.crt**, **server.csr** a **server.key**
- **./build-key-pass client** - certifikát klienta, vytvoří se soubory **client.crt**, **client.csr** a **client.key**
- **./build-dh** -dh parametry, vytvoří se soubor **dh2048.pem**

Nyní máme vytvořeny všechny potřebné certifikáty pro nakonfigurování OpenVPN s TLS zabezpečením.

5.4.5 Konfigurace OpenVPN s TLS

Jak již bylo zmíněno, tato varianta využívá k autentizaci protistran TLS protokol. K tomuto účelu je nutno vygenerovat certifikáty, pomocí kterých bude autentizace probíhat. Tato varianta poskytuje vysokou míru zabezpečení a je téměř neprolomitelná.

Zde jsou konfigurační soubory rozdílné, protože jedna strana plní úlohu serveru a druhá klienta. Nejdůležitější parametry jsou nastaveny v souboru strany serveru (**server.conf**). Parametr **mode** nastaví server mód, **tls** nastaví mód při autentizaci, **ifconfig-pool** nastaví rozsah adres pro přidělení klientům, kde 192.168.5.2 192.168.5.2 znamená možnost pouze jednoho klienta. Dále je nutno zadat cesty k certifikátům a jejich klíčům.

mode server -server mód

tls-server -mód při TLS autentizaci

ifconfig-pool 192.168.5.2 192.168.5.2 255.255.255.0

ca /etc/openvpn/ca.crt	-umístění certifikační autority
cert /etc/openvpn/server.crt	-umístění server certifikátu
key /etc/openvpn/server.key	-umístění soukromého klíče
dh /etc/openvpn/dh2048.pem	-umístění souboru s dh parametry

Na straně klienta (**client.conf**) se parametrem **remote** nastaví IP adresa serveru a **tls** se nastaví klient mód a **pull** vyžádá nastavení serverem. Opět se zadají cesty k certifikátům a klíčům.

remote 192.168.4.1	-IP adresa serveru
tls-client	-mód při TLS autentizaci
pull	-stažení konfigurace ze serveru

Nyní máme nakonfigurované oba počítače a můžeme tunel spustit. Nejdříve spustíme server a následně klienta. Spuštění se provede na obou počítačích příkazem:

/etc/init.d/openvpn start

Na straně klienta budeme navíc dotázaní na heslo, které jsme nastavili při generování klientského certifikátu. V příloze jsou uvedeny konfigurační soubory, detailní postup při tvorbě certifikátů, startovací výpisy serveru i klienta, zachycený hovor útočníkem a nešifrovaný hovor zachycený na rozhranní tap0. Jak již bylo řečeno, útočník je schopen rozeznat pouze IP začátku a konce tunelu, což může být ve firemní síti router tvořící bránu do internetu. Obsah hovoru je útočníkovi neznámý, protože jsou data šifrovaná.

5.5 OpenSWAN

OpenSWAN navazuje na starší projekt FreeS/WAN, jehož vývoj byl zastaven v roce 2004. Dále se vyvíjely už jen projekty OpenSWAN a StrongSWAN. Podpora IPsec je od verze linuxového jádra 2.6 již obsažena v jádru systému. Tuto podporu zajišťuje implementace NETKEY. Pro jádra 2.4 byla používaná implementace KLIPS. Vytváření a správu bezpečnostních asociací zajišťuje IKE démon Pluto, který je součástí OpenSWAN. Lze si ale zvolit i jiného IKE démona (Racoon, Isakmpd).

OpenSWAN jsem instaloval opět v systému Ubuntu 9.10. Instalace proběhne jednoduše příkazem ***apt-get install openswan***. Podobě jako u OpenVPN lze použít několik druhů autentizací obou konců šifrovaného tunelu. Nejjednodušší je použití sdíleného hesla, další

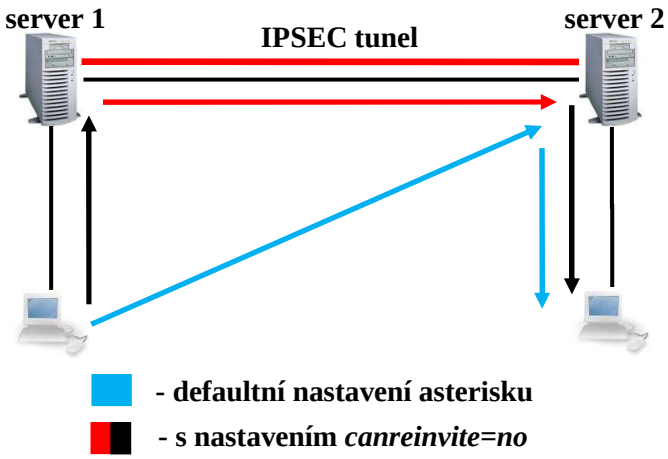
volbou je pomocí asymetrických RSA klíčů a poslední a nejbezpečnější volbou je použití TLS zabezpečení s použitím X.509 uživatelských certifikátů.

Konfigurace OpenSWANu se provádí v souborech **ipsec.conf** - hlavní konfigurační soubor a **ipsec.secrets** - soubor, který obsahuje sdílené heslo v případě autentizace s použitím hesla. Oba dva soubory jsou umístěny v adresáři **/etc**. Soubory v obou počítačích jsou vždy totožné, rozlišení konců tunelu je vyřešeno jako levý konec a pravý konec, kde každému je přiřazena IP adresa počítačů.

U této varianty není nutno, tak jak je tomu u OpenVPN upravovat routovací tabulku v počítačích, protože zde nedojde k vytvoření nové virtuální sítě, do které by bylo nutno přesměrovat síťový provoz.

5.5.1 Úprava nastavení Asterisku v sip.conf

Do souboru **sip.conf** , sekce **[general]** je nutno přidat parametr ***canreinvite=no,*** což znamená, že hovorová data budou směrovány nejdříve na vlastní asterisk server, následné na druhý server a nakonec k příjemci. Defaultně je vlastní server přeskočen a data jsou směrovány rovnou na server příjemce (obrázek 28). Je to důležité z důvodu transportního režimu IPsec tunelu, který je nastaven tak, že šifruje pouze data, která mají zdrojovou a cílovou IP adresu stejnou jako nakonfigurovaný ipsec tunel.



Obrázek 28 cesta RTP paketu podle zdrojové a cílové IP adresy

5.5.2 Konfigurace pomocí sdíleného hesla

Tato varianta se z pohledu bezpečnosti nejeví jako příliš vhodná metoda, autentizace je zajištěna pouze heslem, které si programátor zvolí. Nicméně pro úplnost tuto variantu také uvádím.

Do souboru ***ipsec.secrets*** se zadají konce tunelu a zvolené heslo **192.168.4.1 192.168.4.2: PSK "asterisk"**. Hlavní konfigurace se provede v souboru ***ipsec.conf.*** Parametr **authby** nastavuje typ autentizace, **conn** název šifrovaného tunelu, **left** a **right** IP konců tunelu.

authby=secret - zvolený typ autentizace

conn asterisk - název šifrovaného tunelu

left=192.168.4.1 - IP adresa levého konce tunelu

right=192.168.4.2 - IP adresa pravého konce tunelu

5.5.3 Konfigurace pomocí RSA klíčů

Tato varianta poskytuje podstatně větší zabezpečení, protože autentizace proběhne pomocí dlouhých RSA klíčů, které si každá strana vygeneruje. Konfigurace probíhá pouze v souboru ***ipsec.conf.*** Soubor ***ipsec.secrets*** je v tomto případě použit jako úložiště při generování RSA klíče. Po vygenerování klíčů na obou stranách se oba klíče uloží do ***ipsec.conf***. Tento soubor je opět totožný u obou stran.

K vygenerování klíče použijeme následující příkaz:

ipsec newhostkey -output /etc/ipsec.secrets

RSA klíč přeneseme do ***ipsec.conf*** a doplníme další důležité body nastavení - **authby**, **conn, leftrsasigkey**, **rightrsasigkey -** pravý a levý RSA klíč, **ike** - šifrovací metoda při výměně klíčů, **esp** - šifrovací metoda pro aplikační data

authby=rsasig - autentizace RSA klíčem

conn asteriskRSA - název šifrovaného tunelu

leftrsasigkey=0sAwEAAeyIjCNsoutV6u7F20eUGZ/Hg…….

- RSA klíč levého konce tunelu

rightrsasigkey=0sAwEAAbJ/KpSG9qDeOULbVn15+hauu……

- RSA klíč pravého konce tunelu

ike=aes128-sha1-modp8192 - použité šifrování při výměně klíčů

esp=aes256-sha1 - použité šifrování aplikačních dat

5.5.4 Konfigurace pomocí TLS

Jak již bylo uvedeno, jedná se o nejbezpečnější variantu. Autentizace proběhne pomocí uživatelských certifikátů podepsaných certifikační autoritou. Pro tento účel je nutno opět vygenerovat certifikáty. Lze použít stejný postup jako u vytváření certifikátů pro OpenVPN. Pro konfiguraci OpenSWAN s TLS jsem použil totožné certifikáty, které jsem si vygeneroval při konfiguraci OpenVPN. Do souborů ***ipsec.secrets*** zadám cestu k soukromým klíčům a jejich

hesla **: RSA /etc/ipsec.d/private/server.key "1234"** , v ***ipsec.conf*** je uložena hlavní konfigurace. Tento soubor je pro oba počítače stejný.

Certifikační autoritu **ca.crt** nakopíruju do adresáře: **/etc/ipsec.d/cacerts**

Server.crt a **client.crt** nakopíruju do **/etc/ipsec.d/certs**

Server.key a **klient.key** nakopíruju do **/etc/ipsec.d/private**

V ipsec,conf přidám parametry **leftrsasigkey, rightrsasigkey** - použití certifikátů při autentizaci, **leftcert, rightcert** - jména certifikátů

authby=rsasig

leftrsasigkey=%cert	- autentizace levé strany certifikátem
rightrsasigkey=%cert	- autentizace pravé strany certifikátem
conn asteriskTLS	- název šifrovaného tunelu
leftcert=server.crt	- jméno server certifikátu
rightcert=client.crt	- jméno klientského certifikátu

Spuštění OpenSWAN probíhá následujícími příkazy:

ipsec setup --start	-spuštění openswan
ipsec pluto	-spuštění pluto démona
ipsec auto --add asteriskTLS	-přidání vytvořeného spojení
ipsec auto --ready	-naslouchání druhé strany pro navázání spojení
ipsec auto --up asteriskTLS	-spuštění spojení – tento příkaz se provede pouze na straně serveru

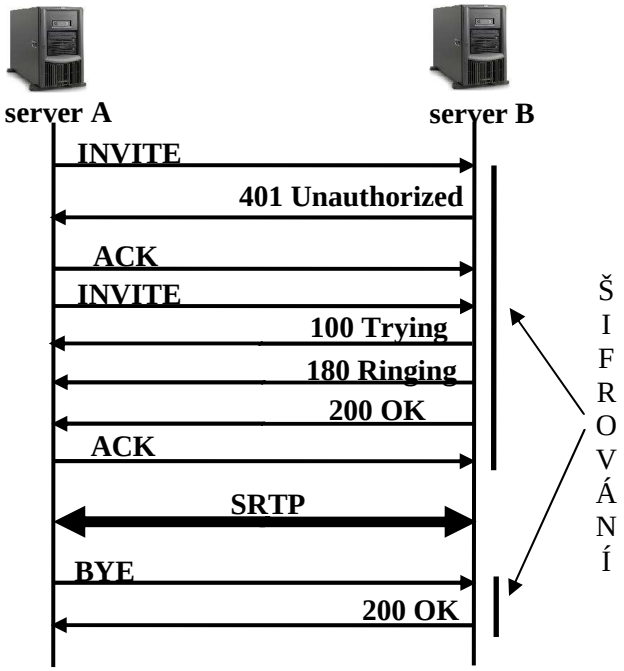
Nyní máme aktivní šifrovaný tunel mezi oběma servery. V příloze jsem uvedl konfigurační soubory, výpis vyjednání spojení, detailní výpis vyjednané asociace pomocí příkazu ***ipsec auto --status*** a zachycený šifrovaný hovor mezi servery. Ze zachyceného hovoru opět vidíme, že nelze určit skutečný zdroj a cíl přenášených dat, jsou vidět pouze IP adresy konců tunelu, obsah dat také nejde rozeznat díky šifrování. Vidíme pouze, že data byla zabezpečena ESP protokolem a číslo SPI, což je pouze identifikátor vyjednané bezpečnostní asociace. SPI lze ověřit z detailního výpisu.

5.6 Zabezpečení signalizace SIPS a médií SRTP

V následujících kapitolách se budu věnovat taktéž kompletnímu zabezpečení hovoru, ale mechanismy k tomu primárně určenými. Zabezpečení bude rozděleno na dvě části - zašifrování signalizace pomocí TLS a hovorových dat SRTP protokolem s AES šifrováním.

5.6.1 Zabezpečení signalizace SIPS

SIPS neboli SIP security je zabezpečení SIP protokolu proti útoku na SIP signalizaci pomocí TLS. Komunikace probíhá na portu 5061. Předchozí zabezpečení by si vystačily s asteriskem 1.4, pro toto řešení (bez SRTP) je nutno použít kteroukoli verzi Asterisku od verze 1.6, která již podporuje TLS. Komunikující strany jsou vzájemně autentizovány a SIP zprávy a odpovědi jsou zašifrovány. Jedinou nevýhodou je, že přenos RTP dat zůstane nešifrován a je nutno použít další metodu zabezpečení, například SRTP (obrázek 29). Tato volba bude obsažena až v Asterisku ve verzi 1.6.3. Při vyjednávání TLS spojení mezi dvěma Asterisk servery není pevně dáno, kdo bude plnit úlohu serveru a kdo klienta. TLS vyjednání neproběhne při startu Asterisku, ale až při spojování hovoru při použití trunku. Úlohu klienta bude plnit ten Asterisk, který bude žádat druhý Asterisk o spojení hovoru. Takže pokud bude klient z PC3 chtít volat klienta v PC4, bude Asterisk server1 (PC1) plnit úlohu klienta při TLS vyjednávání.



Obrázek 29 šifrování SIP zpráv a hovorových dat

Konfigurační soubory asterisku s TLS jsou totožné jako konfigurace použitá u předchozích řešení, pouze do **sip.conf** se přidá několik řádků pro TLS nastavení. Je nutno vygenerovat uživatelské certifikáty, opět bude použit podobný postup jako u vytváření certifikátů pro OpenVPN.

5.6.2 Certifikáty pro Asterisk 1.6

Při generování certifikátu asterisk1 je nutno do pole **common name** zadat IP adresu asterisk serveru1 (192.168.4.1) a do certifikátu asterisk2 zadat IP adresu asterisk serveru2 (192.168.4.2). Generování proběhne následujícími příkazy:

. ./vars
./clean-all
./build-ca
./build-key asterisk1
./build-key asterisk2

Budou vytvořeny soubory **ca.crt**, **ca.key**, **asterisk1.crt**, **asterisk1.key**, **asterisk2.crt** a **asterisk2.key**.

Dále je nutno spojit certifikáty se svými klíči:

cat asterisk1.crt asterisk1.key > asterik1key.pem

cat asterisk2.crt asterisk2.key > asterik2key.pem

Alternativním řešením tvorby uživatelských certifikátů je použití OpenSSL balíčku. V souboru openssl.cnf se nastaví parametry pro certifikáty. Certifikační autorita se vygeneruje příkazem ***openssl req -new -x509 -nodes –out ca.crt -keyout ca.key -days 3650***, opět jsme vyzváni k zadání několika identifikačních údajů a hesla. Budou vytvořeny soubory **ca.crt** a **ca.key**. Dále vytvoříme zatím ještě nepodepsaný certifikát a jeho klíč ***openssl req -new -nodes -out asterisk1.pem -keyout asterisk1.key -days 3650***. Budou vytvořeny soubory asterisk1.pem a asterisk1.key. Podepsání certifikátu certifikační autoritou provedeme příkazem ***openssl ca -in asterisk1.pem -out asterisk1.crt.*** Obdobně vygeneruji certifikát pro asterisk 2. Opět je nutno spojit certifikáty se svými klíči podle výše uvedeného postupu.

Na obou serverech jsem vytvořil adresář **/etc/asterisk/CA** do kterého jsem nakopíroval certifikační autoritu **ca.crt** a do adresářů **/etc/asterisk/C** jsem nakopíroval soubory **asterisk1key.pem** a **asterisk2key.pem**, každý pouze na svůj server.

V **sip.conf** do sekce [general] se přidají cesty k certifikátům, parametr **tlsenable** a u nastavení trunku se přidají parametry **port** a **transport.**

[general]	
tlsenable=yes	- podpora TLS
tlscertfile=/etc/asterisk/C/asterisk1key.pem	- cesta k certifikátu
tlscafile=/etc/asterisk/CA/ca.crt	- cesta k certifikační autoritě
tlscadir=/etc/asterisk/CA	- adresář umístění certifikačních autorit
[trunk1]	
port=5061	- port pro SIPS
transport=tls	- tls spojení

5.6.3 Zabezpečení médií SRTP

Jak již bylo zmíněno, zabezpečení SIPS poskytuje ochranu pouze SIP zprávám a hovorová data zůstávají nechráněna. Proto je nutno místo RTP použít pro přenos hovorových dat SRTP. Data jsou šifrovány AES šifrováním a hashováním SHA1. Zachycený SRTP provoz je zobrazen na obrázku 30. Požadavek na šifrovací přenos je zaslán v INVITE zprávě. SRTP šifrování dosud není obsaženo v žádné verzi Asterisku, ale měla by se objevit ve verzi 1.6.3. Proto je nutno SRTP podporu doinstalovat v patch balíčku nebo si stáhnou jednu z neoficiálních verzí Asterisku obsahující SRTP. V tomto případě musíme počítat s kompilací Asterisku ze zdrojového kódu. Postup kompilace Asterisku je popsán v příloze.

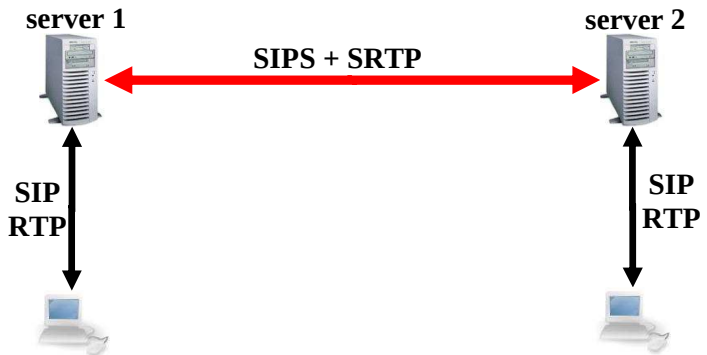
```
24 2.419080   192.168.4.1  192.168.4.2  SRTP   PT=ITU-T G.711 PCMA, SSRC=0x4A6CFB99, Seq=26777, Time=18952
25 2.427645   192.168.4.2  192.168.4.1  SRTP   PT=ITU-T G.711 PCMA, SSRC=0x5D2B9702, Seq=65078, Time=13125
26 2.439082   192.168.4.1  192.168.4.2  SRTP   PT=ITU-T G.711 PCMA, SSRC=0x4A6CFB99, Seq=26778, Time=18954
27 2.445469   192.168.4.2  192.168.4.1  SRTP   PT=ITU-T G.711 PCMA, SSRC=0x5D2B9702, Seq=65079, Time=13127
```

Obrázek 30 Zachycené SRTP pakety

Opět je nutno do konfiguračních souborů sip.conf a extensions.conf přidat řádky povolující SRTP šifrování. V sip.conf se do trunku přidá **srtpcapable=yes** a v extensions.conf se nad nastavení trunku přidá **exten=>_2.,1,Set(_SIPSRTP=1).** Dále je nutno v dalších řádcích nastavení trunku zvýšit číslo priority o 1.

exten=>_2.,1,Set……. → **exten=>_2.,2,Set**…….
exten=>_2.,2,Dial…… → **exten=>_2.,3,Dial**……

Nyní máme nastavenou kompletní ochranu hovoru. Po spuštění obou Asterisků a zprostředkování hovoru, budou SIP zprávy i hovorová data mezi servery šifrovány (obrázek 31). Do přílohy jsem přidal konfigurační soubory, zachycený zašifrovaný hovor a jeho dešifrovanou podobu. Vidíme, že všechna data jsou opravdu šifrovaná. V programu Wireshark lze SIP zprávy dešifrovat, je nutno mít soukromý klíč serveru a v nabídce **edit/preferences/ssl** do pole **RSA key list** zadat IP, port, protokol a cestu ke klíči. V mém případě je pole vyplněno následovně: **192.168.4.1,5061,sip,c:\documents and settings\zbynek\plocha\c\asterisk1.key** Nyní už rozeznáme všechny SIP zprávy, kterými mezi sebou servery komunikovaly a vidíme, že hovor je chráněn SRTP zabezpečením.



Obrázek 31 Šifrování hovoru mezi servery

6 Porovnání použitých řešení

6.1 Obtížnost implementace

V porovnání všech implementací na úrovni instalace vyjdou OpenSWAN a OpenVPN jako složitější na instalaci a zprovoznění, z hlediska funkčnosti jsem neshledal žádný problém. Všechny pracovaly správně a podle všech předpokladů a nároků, které byly kladeny na funkčnost a součinnost s VoIP řešeními. Vždy zde byla možnost konfigurace tak, aby se OpenVPN i OpenSWAN spustily hned po startu systému a automaticky se vyjednaly bezpečnostní asociace a vytvořil se šifrovaný tunel, aby uživatel nemusel vždy po startu systému tyto operace provádět znovu. Všechny řešení vyžadovaly vytvoření certifikátů, pro zvýšení bezpečnosti, pomocí autentizace obou stran spojení.

Implementace OpenVPN se pohodlně konfiguruje v jednom souboru s příponou **.conf** a to jak na straně serveru, tak klienta. Konfigurační soubory serveru a klienta v režimu TLS nemají stejnou podobu. Pro tento režim je nutno vygenerovat certifikáty, k čemuž byly použity skripty pro generování certifikátů. Zabezpečení VoIP telefonie pomocí OpenVPN vyžaduje úpravu souboru **sip.conf** v asterisku, kde se musí změnit IP adresa hosta na adresu nově vytvořeného tunelu. Stejně tak se musí změnit routovací tabulka, aby byla pro komunikaci použita nová virtuální síť.

U OpenSWAN probíhá konfigurace také bez problémů v jednom souboru **ipsec.conf**. Opět je nutno vytvořit certifikáty, postup je totožný jako u OpenVPN proto je možno použít certifikátů, které byly vytvořeny pro OpenVPN. Routovací tabulka se zde neupravuje, pouze je nutno upravit asterisk v **sip.conf** tak, aby cesta hovorových dat probíhala přes oba asterisk servery.

Řešení bezpečnosti IP telefonie pomocí SIPS a SRTP je z hlediska konfigurace asi nejjednodušší, pomineme-li nutnost kompilace Asterisku, tato možnost ale zanedlouho odpadne počínaje verzí Asterisku 1.6.3. Stačí vygenerovat certifikáty podobně jako u předchozích řešení a do souboru **sip.conf** v Asterisku přidat několik řádků, pro TLS a SRTP konfiguraci a cestu k certifikátům. Tento krátký postup stačí pro kompletní zabezpečení hovoru.

6.2 Odolnost proti útoku

Při porovnání bezpečnosti všech řešení nám vyjde nejlépe OpenSWAN a OpenVPN. Obě implementace byly vyvinuty pro zabezpečení jakéhokoli provozu v IP síti, ale každá to provádí odlišným způsobem. Útočník ze zachycených dat nedokáže rozeznat, že jde o telefonní hovor. Poskytují autentizaci obou stran a zajišťují kvalitní šifrování šifrou AES, která dodnes ještě nebyla prolomena. Zabezpečení SIPS v kombinaci s SRTP poskytne taktéž kompletní ochranu hovoru, útočník však rozezná, o jaký typ dat se jedná. Nicméně více informací ze zachycených dat nezíská. SIPS je schopno ochránit hovor proti útoku na signalizační zprávy,

SRTP pomocí AES šifrování zajistí ochranu proti odposlechu. Vyjednání klíčů pro SRTP je zde chráněno v zašifrovaných SIP zprávách. Pokud by nebylo použito TLS na SIP zprávy, bylo by nutno pro vyjednání klíčů použít některý z dalších zabezpečovacích mechanismů, například ZRTP.

6.3 Nárůst datového toku mezi servery

Při implementaci OpenVPN a OpenSWAN na VoIP telefonii jsem měřil, o kolik se zvětší datový tok mezi Asterisk servery, kde je datový provoz zabezpečen. Výsledky jsou uvedeny v tabulkách 1 a 2.

Byly uskutečněny hovory mezi ústřednami propojenými pomocí SIP trunku a IAX trunku, ten byl navržen vývojáři Asterisku pro snížení datové zátěže mezi Asterisk servery, proto bylo porovnáno, o kolik se sníží datový tok použitím IAX trunku. Hovory byly uskutečněny ze softwarových klientů X-lite a byly použity dva typy kodeků. Kodek G.711 s bitovou rychlostí 64 kbps a dále byl vybrán kodek GSM s nízkou bitou rychlostí 13 kbps. Nejdříve byly všechny hovory uskutečněny přes nezabezpečenou síť a následně byla síť zabezpečena postupně pomocí OpenSWAN a OpenVPN s šifrováním AES 256.

Datový tok [kbps]	G.711		GSM	
	SIP	IAX	SIP	IAX
bez šifrování	90	82	39	24
IPsec/OpenSWAN	121	111	66	60
VPN/OpenVPN	128	121	76	70

tabulka 1: měření datového toku

Naměřené hodnoty jsou zapsány v tabulce 1 a zaneseny do grafů č.1 a 2 v příloze. Z naměřených dat vidíme, že datový tok při jednom hovoru s použitím G.711 kodeku spojený SIP trunkem je 90 kbps, při použití IAX trunku se datový tok sníží na 82 kbps, což znamená úsporu asi 12 %. U GSM kodeku je IAX úspornější, pokles činí zhruba 38 %. Při zabezpečení serverů pomocí OpenSWAN vzroste datový tok jednoho hovoru s kodekem G.711 o 35 % (SIP i IAX) a u kodeku GSM je nárůst o 69 % - SIP a 150 % - IAX. Při zabezpečení hovoru pomocí OpenVPN bylo naměřeno navýšení datového toku ještě výraznější. Nejmenší navýšení bylo opět u hovoru spojeného kodekem G.711 42 % - SIP a 47 % - IAX. Použitím kodeku GSM se navýšil datový tok o 94 % - SIP a 191 % - IAX.

Pro další měření byl použit software IxChariot, který slouží ke generování telefonních hovorů. Měřil jsem datový tok při jednom hovoru generovaným tímto softwarem a při 50-ti

hovorech. Opět bylo měřeno spojení přes síť bez šifrování a s šifrováním. U OpenVPN jsem v tomto měření nastavil ještě LZO komprimaci dat, abych porovnal, jestli se datový tok nějak výrazně sníží. Délky zachycených paketů ale měly shodnou délku, takže pokles byl nulový. Kodeky byly použity opět G.711 a jako nízkorychlostní kodek G.729 s bitovou rychlostí 8 kbps.

Datový tok [kbps]	G.711		G.729	
	1 hovor	50 hovorů	1 hovor	5O hovorů
bez šifrování	90	3959	31	1363
IPsec/OpenSWAN	117	5162	56	2472
VPN/OpenVPN (bez komprimace)	128	5642	67	2939
VPN/OpenVPN (s komprimací)	128	5642	67	2939

tabulka 2: měření datového toku (IxChariot)

Naměřené hodnoty jsou zapsány v tabulce 2 a zaneseny do grafů č.3,4,5 a 6 v příloze. Při porovnání hovoru zabezpečeným OpenVPN vidíme, že použití LZO komprimace nemá vliv na snížení datového toku. Při 50-ti hovorech s G.711 vzroste datový tok u OpenSWAN o 30 %, u OpenVPN o 42 %. S použitím kodeku G.729 je nárůst u OpenSWAN o 80 %, u OpenVPN o 116 %. Z naměřených dat vidíme, že šifrování hovorů spojených nízkorychlostními kodeky přináší podstatně větší nárůst datového toku než u hovoru s vysokou bitovou rychlostí.

U zabezpečení hovoru SRTP protokolem je nárůst datového toku u G.711 asi 4 % a u GSM 12 %. Datový paket je zde zvětšen o 10 B autentication header HMAC.

7 Závěr

Cílem této diplomové práce bylo popsat mechanismy zabezpečení IP sítě, která je využívaná pro VoIP telefonii a realizovat IP síť zabezpečenou pomocí vybraných implementací OpenSWAN, OpenVPN, SIPS a SRTP. Jelikož se v několika dalších letech počítá s rapidním nárůstem využití VoIP telefonie, je nutno stále více myslet na bezpečnost takového hovoru. Byly popsány mechanismy pro zabezpečení jak celé IP sítě, tak pouze pro technologii VoIP, konkrétně zabezpečení signalizačních SIP zpráv mezi dvěma Asterisk servery. Aby si čtenář mohl udělat představu o fungování celé VoIP sítě, byla tato problematika popsaná od základního RTP protokolu, přes SIP protokol, až po zabezpečovací mechanismy, používané jako ochrany před útoky na IP síť a VoIP telefonii.

V další části jsem popisoval, jakými způsoby lze zajistit autentizaci v SIP protokolu při registraci klienta k serveru a při sestavování spojení. S autentizací souvisí bezpečnost IP telefonie, kdy je důležité zpřístupnění služeb pouze povoleným klientům. Bylo popsáno rozdělení autentizačních mechanismů do tří vrstev podle modelu RM-OSI.

Pro realizaci praktické části diplomové práce jsem použil 5 fyzických počítačů, pro simulaci reálných podmínek v síti. Byly vytvořeny tři IP sítě, kde dvě simulovaly například podnikovou síť, které byly propojeny přes třetí síť, simulující internetovou síť, odkud hrozí největší riziko napadení útočníkem. Právě na tuto část jsem realizoval bezpečností mechanismy. Byly použity implementace OpenSWAN a OpenVPN, které pomocí SSL/TLS zabezpečení a šifrování AES zaručí bezpečnou komunikaci mezi volajícími stranami. Obě varianty řeší zabezpečení odlišným způsobem, ale výsledná míra zabezpečení je na nejvyšší úrovni. Při zabezpečení IP sítě těmito implementacemi došlo k nárůstu datového provozu na zabezpečené části z důvodu šifrování dat. Nejmenší nárůst činil u hovoru s kodekem s vysokou bitovou rychlostí propojeným SIP trunkem, naopak nejvyšší nárůst byl zaznamenán při hovoru s nízkorychlostním kodekem propojeným IAX trunkem. Alternativou k těmto řešení je kombinace zabezpečení SIPS a SRTP, které zašifrují SIP signalizaci pomocí TLS a hovorová data AES šifrováním. Při použití SRTP šifrování bude nárůst činit 4 a 12 % (G.711, GSM).

Při praktické realizaci zabezpečení pomocí tunelu s IPsec a TLS jsem se rozhodl pro použití open-source nástrojů v linuxovém prostředí pomocí aplikací OpenSWAN a OpenVPN, a to vzhledem k dostupnosti, vysoké vyspělosti, otevřenosti kódu a dokumentaci. V práci jsem poskytl veškeré podklady, které umožňují reprodukovatelnost použitého řešení a ověření mnou dosažených výsledků.

Literatura

[1] VOZŇÁK, Miroslav. *Voice over IP*. Skripta VŠB. Ostrava : VŠB, 2008. 176 s. ISBN 978-80-248-1828-3

[2] *TCP/IP : kompletní průvodce*. Praha : Soft Press s.r.o., 2002. 512 s. ISBN 80-86497-34-8

[3] SINNREICH, Henry. *SIP Beyond VoIP : The Next Step in the IP Communications Revolution* . New York : VON Publishing LLC, 2005. 309 s. ISBN 0-9748130-0-1

[4] KUHN, D. Richard; WALSH, Thomas J.; FRIES, Steffen. *Security Considerations for Voice Over IP Systems*. Gaithersburg : National Institute of Standards and Technology, 2005. 93 s. Dostupné z WWW: <http://csrc.nist.gov/publications/nistpubs/800-58/SP800-58-final.pdf>

[5] RANSOME, James; RITTINGHOUSE, John. *VoIP Security*. Oxford : Elsevier, 2005. 402 s. ISBN 1-55558-332-6

[6] MEGGELEN, Jim V.; SMITH, Jared; MADSEN, Leif. *Asterisk : The Future of Telephony*. Sebastopol : OReily, 2005. 380 s. ISBN 0-596-00962-3

[7] *Asterisk* [online]. 2009 [cit. 2010-04-20]. The Open Source Telephony Project. Dostupné z WWW: <http://www.asterisk.org/>

[8] *Cesnet* [online]. 2007 [cit. 2010-04-20]. Sip. Dostupné z WWW: <https://sip.cesnet.cz/cs/protokoly/sip>

[9] VESTERINEN, Pauli. *Helsinki University of Technology* [online]. Helsinki : 2006 [cit. 2010-04-20]. User authentication in SIP. Dostupné z WWW: <http://www.tml.tkk.fi/Publications/C/22/papers/Vesterinen_final.pdf>

[10] REPIQUET, Joël. *Tech-invite* [online]. 2005 [cit. 2010-04-20]. SIP. Dostupné z WWW: <http://www.tech-invite.com/Ti-sip-ex3261.html>

[11] DOSTÁLEK, Libor. *Cpress* [online]. 1997 [cit. 2010-04-27]. S/MIME. Dostupné z WWW: <http://www.cpress.cz/knihy/tcp-ip-bezp/CD-pem/smime4.htm>

[12] MACHNÍK, Petr. *Přednášky - Širokopásmové sítě* [online]. Ostrava : VŠB, 2009 [cit. 2010-05-02]. Dostupné z WWW: <http://moodle.kat440.vsb.cz/>

[13] *Openvpn* [online]. 2002 [cit. 2010-04-20]. OPENVPN - SECURE YOUR CONNECTIVITY. Dostupné z WWW: <http://openvpn.net/>

[14] *Wikipedia* [online]. 2010 [cit. 2010-05-02]. Virtual private network. Dostupné z WWW: <http://en.wikipedia.org/wiki/Virtual_private_network>

[15] *Abclinuxu* [online]. 2007 [cit. 2010-04-20]. Debian Etch - OpenVPN klient/server. Dostupné z WWW: <http://www.abclinuxu.cz/blog/bl4z4/2007/6/7/182859>

[16] HLADÍK, Radek. *ROOT.CZ* [online]. 2004 [cit. 2010-04-20]. OpenVPN - VPN jednoduše. Dostupné z WWW: <http://www.root.cz/clanky/openvpn-vpn-jednoduse/>

[17] *Openswan.org* [online]. 2003 [cit. 2010-04-20]. OpenSWAN. Dostupné z WWW: <http://www.openswan.org/>

[18] *The official IPsec Howto for Linux* [online]. 2007 [cit. 2010-04-20]. IPsec. Dostupné z WWW: <http://www.ipsec-howto.org/>

[19] *Wikipedia* [online]. 2010 [cit. 2010-04-20]. IPsec. Dostupné z WWW: <http://en.wikipedia.org/wiki/IPsec>

[20] KÁRA, Michal. *ROOT.CZ* [online]. 2003 [cit. 2010-04-20]. Tuneluji, tuneluješ, tunelujeme: IPSec. Dostupné z WWW: <http://www.root.cz/clanky/tuneluji-tunelujes-tunelujeme-ipsec/>

[21] *VoIP-info* [online]. 2008 [cit. 2010-04-27]. Asterisk Documentation 1.6.0 siptls.txt. Dostupné z WWW: <http://www.voip-info.org/wiki/view/Asterisk+Documentation+1.6.0+siptls.txt>

[22] *VoIP-info* [online]. 2009 [cit. 2010-04-27]. Asterisk SRTP. Dostupné z WWW: <http://www.voip-info.org/wiki/view/Asterisk+SRTP>

[23] *Lists.digium* [online]. 2009 [cit. 2010-04-27]. SRTP and Dialplan control. Dostupné z WWW: <http://lists.digium.com/pipermail/asterisk-dev/2009-January/036029.html>

[24] SCHULZRINNE, H, et al. *The Internet Engineering Task Force* [online]. 1996 [cit. 2010-04-29]. RTP: A Transport Protocol for Real-Time Applications. Dostupné z WWW: <http://www.ietf.org/rfc/rfc3550.txt>

[25] BAUGHER, M, et al. *The Internet Engineering Task Force* [online]. 2004 [cit. 2010-04-27]. The Secure Real-time Transport Protocol. Dostupné z WWW: <http://www.ietf.org/rfc/rfc3711.txt>

[26] ROSENBERG, J, et al. *The Internet Engineering Task Force* [online]. 2002 [cit. 2010-04-27]. Session Initiation Protocol. Dostupné z WWW: <http://www.ietf.org/rfc/rfc3261.txt>

[27] CHOWN, P. *The Internet Engineering Task Force* [online]. 2002 [cit. 2010-04-27]. Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS). Dostupné z WWW: <http://www.ietf.org/rfc/rfc3268.txt>

[28] DIERKS, T. *IETF Tools* [online]. 2008 [cit. 2010-04-27]. The Transport Layer Security (TLS) Protocol Version 1.2. Dostupné z WWW: <http://tools.ietf.org/html/rfc5246>

[29] FRANKS, J, et al. *The Internet Engineering Task Force : HTTP Authentication: Basic and Digest Access Authentication* [online]. 1999 [cit. 2010-04-20]. RFC 2617. Dostupné z WWW: <http://www.ietf.org/rfc/rfc2617.txt>

[30] KENT, S; SEO, K. *The Internet Engineering Task Force : Security Architecture for the Internet Protocol* [online]. 2005 [cit. 2010-04-20]. Rfc4301. Dostupné z WWW: <http://tools.ietf.org/html/rfc4301>

[31] KENT, S. *The Internet Engineering Task Force : IP Authentication Header* [online]. 2005 [cit. 2010-04-20]. Rfc4302. Dostupné z WWW: <http://tools.ietf.org/html/rfc4302>

[32] KENT, S. *The Internet Engineering Task Force : IP Encapsulating Security Payload (ESP)* [online]. 2005 [cit. 2010-04-20]. Rfc4303. Dostupné z WWW: <http://tools.ietf.org/html/rfc4303>

Přílohy

Seznam příloh:

1 OpenVPN

Konfigurační soubory pro TLS zabezpečení

Startovací výpis server

Startovací výpis klient

Zachycený hovor na rozhranní eth0

Zachycený hovor na rozhranní tap0

Generování certifikátů pro OpenVPN a OpenSWAN

2 OpenSWAN

Konfigurační soubory pro TLS zabezpečení:

Start OpenSWAN

Podrobný výpis spojení sestaveného tunelu na straně serveru:

Výstavba spojení

Zachycený šifrovaný hovor

3 SIPS + SRTP

Kompilace Asterisku s podporou SRTP v systému Ubuntu 9.10

Konfigurační soubory

TLS handshake a následné šifrování SIP signalizace a hovorových dat

Děšifrovaná SIP signalizace

4 Grafy naměřených hodnot

Přílohy na CD:

Certifikáty

Konfigurační soubory

Zachycené hovory